System Control and Intelligent Protection in a DC Traction Substation

E. Beaud

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering at the University of Stellenbosch

Supervisor : Prof. J.H.R. Enslin

December 1997

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, unless otherwise stated, and has not previously, in its entirety or in part, been submitted at any university for a degree.

E. Beaud

December 1997

Summary

A prototype dc traction substation has been developed that consists of several subsystems. The system models the operation of a traction substation and simulates a traction load. Two additions are proposed to the substation, the one being an anti-parallel path for active power filtering and regeneration. The other is a high voltage IGBT dc switch developed by the author to dump excessive regenerated energy into resistor banks. PI and fuzzy logic control algorithms have been implemented to control the dc dump. The operation of the controllers were practically evaluated and a comparison drawn. The protection of the system was then expanded from the localised dc dump protection to system wide protection control. The concept of "intelligent" control was investigated and implemented in two system protection controllers. A shutdown protection strategy was put in juxtaposition to a fuzzy logic controller in the protection of the system. Both system controllers were practically evaluated and a comparison is drawn between the two. Some conclusions are made concerning the protection issues addressed in the thesis.

Opsomming

'n Prototipe gelykstroom (gs) traksie stelsel is ontwikkel wat uit verskeie sub-stelsels bestaan. Die stelsel modelleer die werking van 'n traksie substasie en simuleer 'n traksie las. Daar word twee byvoegings tot die huidige stelsel voorgestel. Die een is 'n anti-parallel pad vir aktiewe drywingsfiltering en regeneratiewe energie. Die ander is 'n hoogspanning IGBT gs skakelaar wat deur die outeur ontwikkel is om oormatige regeneratiewe energie in resistor banke te verkwis. Beide PI en "Fuzzy Logic" beheerders is geïmplementeer om die gs skakelaar te beheer. Die werking van beide beheerders is prakties ondersoek en 'n vergelyking word getref. Die beveiliging van die stelsel word dan uitgebrei van die gelokaliseerde beveiliging van die gs skakelaar tot stelselwye beveiligingsbeheer. Die konsep van "intelligente" beheer is ondersoek en op twee wyses geïmplementeer. 'n Aan/af beveiligings strategie is afgespeel teen 'n "Fuzzy Logic" beheerder in die beveiliging van die stelsel. Beide stelsel beheerders is prakties geëvalueer en die twee is met mekaar vergelyk. Sekere afleidings word gemaak na aanleiding van die beveiligingskwessies wat in die tesis aangeraak word.

Acknowledgements

I would like to thank the following people, institutions and person :

Prof. J.H.R. Enslin for the opportunity, the wisdom and help. For being the drive behind it all.

Mr. Pietro Petzer and the workshop staff for their patience with me and their willingness to help.

My colleagues : Leonardo Pittorino, Wernher Swiegers, Richardt Wilkinson, Aniel le Roux, Carlo Putter, Braam Horn, et. al., for continued friendship, support and generous input.

Spoornet SA for funding the project and supplying certain critical data.

FRD for generous funding

Jesus Christ - for being the rock on which I stand (Psalm 91:2)

TABLE OF CONTENTS

List of Figures

List of Tables

List of Symbols and Abbreviations

/div	Per division
3ϕ	Three phase
A/D	Analogue to Digital
AC	Alternating current
AHDL	Altera™ Hardware Development Language
APF	Active Power Filter
Aux.	Auxiliary
C/T	Current Transformer
CCC	Central Control Computer
COM	Centre of Maximum
CRDM	Current Regulated Delta Modulation
D/A	Digital to Analogue
DC	Direct Current
DCM	DC Motor
DOM	Degree of Membership
DSP	Digital Signal Processor
EL	Earth Leakage
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EPLD	Enhanced/Erasable Programmable Logic Devise
ESCOM	Electricity Supply Company
FACTS	Flexible AC Transmission Systems
GTO	Gate Turn-off Thyristor
H.C.	High-speed Circuit breaker
HV	High Voltage
IEE	Institute of Electrical Engineers
IEEE	Institute of Electrical and Electronic Engineers
IGBT	Insulated Gate Bipolar Transistor
LC	Inductance - Capacitance
Lq	Line Inductance

MG Set	Motor Generator Set
MOV	Metal Oxide Varistor
NRS	National Regulator Standard
O/L	Over load
PC	Personal Computer
PI	Proportional Integral
PWM	Pulse Width Modulation
rad	radians
RC	Resistor - Capacitor
SAIEE	South African Institute of Electrical Engineers
SM	Synchronous Motor
T/B	Track Breaker
THD	Total Harmonic Distortion
U.V.P.	Under - Voltage Protection
VAR	Volt Ampere Reactive
W.F. Equip.	WaveForm Equipment

Chapter 1

Introduction

1. Introduction

1.1 Background

The dc traction system of South Africa consists of three main electrical components.

- A network of rectifier substations
- A collection of rolling stock, i.e. locomotives.
- The overhead track equipment

The ac network feeding the substations connects directly to the utility grid at specific points where metering is done to determine Spoornet's electricity account. The power is then fed through overhead lines to substations placed at intervals from 8 km to 20 km along the railway lines. The dc overhead line is divided into matching sections, each powered from both ends by a traction substation. The ac input to the different substations varies from 11 kV to 132 kV The transformers convert the ac voltage to two 1.2 kV line to line secondaries that feed a standard twelve pulse rectifier. The substation houses the rectifier, the necessary switch gear and protection as well as switches for regenerative energy in some cases [M1]. The regenerative resistor banks are placed outside the substation buildings. The rectifier powers the overhead 3.3 kV dc lines that feed the traction motors of the locomotives. A block diagram of the Spoornet system is shown in Figure 1-1.

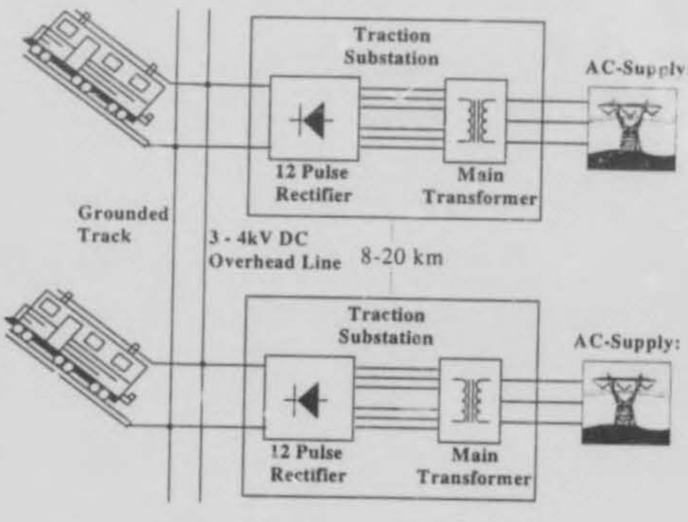


Figure 1-1: Spoornet traction system

The system can be described to operate in two modes of operation. The first is the powering mode in which power flows from the ac grid through the rectifier to the dc side powering the traction load. The amount of power flow in the forward direction is typically in the order of 4 to 6 MVA (Figure 1-2). The other mode of operation is when the driver uses the traction motors of the locomotive to brake the train. This regenerates power which flows back to the substation to be dissipated. More will be said about this in the next section.

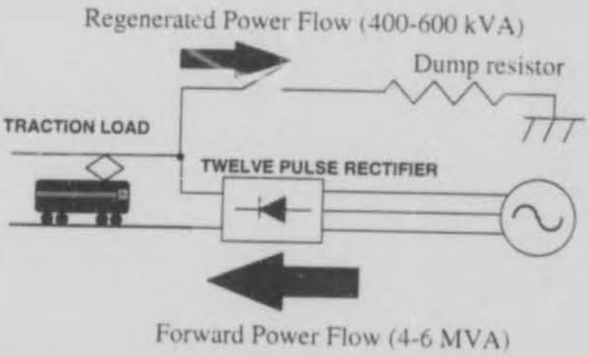


Figure 1-2: Traction system modes of operation

The amount of regenerated power differs greatly with factors such as location and track layout. A study has been done on the subject and approximately 10 % [M2] of the forward path power flow will be regenerated on average. The Kobe Municipal Transportation Bureau have employed regenerative substations that use a parallel path of power flow to inject regenerated energy back into the ac supply [A1] .

1.2 Braking Strategies

The driving of the locomotives in the current set-up relies heavily on the skill of the driver. The driver can select different "notches" of acceleration for the traction engines which relates to discrete steps in torque applied. In the case where the train needs to decelerate for a curve in the track or for a stoppage, the driver has two options. As mentioned earlier the driver can use the traction motors to act as generators. This then uses the kinetic energy stored in the momentum of the train to generate electric energy. In the present system this energy can flow via the overhead line to another locomotive that is in the same section at the time [A2] . If that train is in the powering mode it can use the regenerated energy instead of drawing power from the utility. If there is not another train in the section at the time the power can not flow back through the traction rectifiers as they can only operate in the first quadrant.

The substations that feed sections where regeneration [A3] is common are equipped with large resistor banks. These are switched onto the dc bus by 100 A dc contactors during regeneration. At 3.3 kV this constitutes a regenerative capability of 330 kW per resistor bank. At some substations there are up to nine resistor banks giving a total regenerative capability of 3 MW. If the total regeneration of the traction motors is more than the regenerative capability of the substation or the regeneration system fails, a backup alternative is needed.

The other option for the driver is the mechanical brakes that are installed in all the cars in the train. The driver can apply these brakes in addition to the traction motors or as a backup for failure of the regeneration system. Some of the comparative benefits and liabilities are summarised in Table 1-1.

Table 1-1: Mechanical vs. electrical braking

	Regenerative Braking	Mechanical Braking
Cost	Save on Electricity	Parts Expensive
Effectiveness	Limited by Substation Capability	Limited by wheel slip on whole train
Reliability	Depends on switch gear and control	Depends on mechanical parts
Wheel slip	Braking distributed over locomotive wheels	Braking distributed over all the wheels of the train
Smoothness	Nine discrete steps	Old trains not as smooth as the electrical brakes

From the above table it can be seen that in the areas of effectiveness, reliability and wheel slip the mechanical brakes are better. On cost and smoothness of operation the electrical braking method is better. It is hard to say that one method is better than the other, but it can be said that the combination of the two methods are optimal. The reliability of the mechanical braking system is essential and the additional costs of an electrical braking system would be compensated for both in the saving of electricity as well as the saving in the brake pads of the mechanical brakes.

1.3 Harmonic Compensation Techniques in Traction Substations

A severe problem that Spoornet is facing at the moment is the harmonic content of the currents drawn by the twelve pulse rectifiers. Internationally utilities have set standards for the harmonics drawn by their consumers. Some international standards are shown in ([A4] to [A8], [T1]). The South African utility ESKOM has drafted a standards document, the NRS048 ([A9], [M3], [M4]) that sets the limits on harmonics drawn by the end user. The total THD allowed on the voltage is 8 %. The 5^{th}, 7^{th}, 11^{th} and 13^{th} current harmonics are limited to 6 %, 5 %, 3.5 % and 3 % respectively. The Spoornet substations do not adhere to the limits in the current system (A preliminary study shows a voltage THD of about 10 % [T2] - this does depend on the impedance of the ac grid). A further problem that is being experienced is that the dc side harmonics of the 14E locomotive interfere with the signalling [T3].

An additional problem that the traction substation poses is the power factor of the currents drawn. The power factor of the dc substations is very close to unity, but if the compensation technique for harmonics could be combined with that for power factor then equipment would be utilised maximally. Modern systems usually employ static VAR compensators to rectify the power factor [T4], but no compensation is currently in place in the Spoornet traction substations. Some suggestions have been made on what compensation techniques would be the most effective ([A10] to [A10][A13]) and the solutions that were offered were passive LC filters and/or an IGBT passive filter ([B1] to [B3]). A short summary of the comparison of the two solutions is shown in

Table 1-2.

Table 1-2: Active power filtering vs. passive filtering

	Active power filtering	Passive filtering
Cost	Expensive switching elements	Expensive passive elements
Maintenance	Needs highly qualified personnel	Needs regular tuning
Response	Real time calculation	Deteriorates with time
Regeneration	Energy is injected into the AC side	Energy is dissipated in resistors
Discerning	Only compensates for one substation's harmonics	Filter needs to be rated for all harmonics on the ac side
Technology	New and developing	Well developed

1.4 The Prototype System

A 10 kVA prototype active power filter has been developed and tested by Peter-Jan Randewijk [T2] at the University of Stellenbosch. The filtering action was combined with regeneration by placing the active power filter in anti-parallel with the traction rectifier [M5]. A higher power prototype system [T5] has been developed at the University of Stellenbosch to operate at actual practical voltage levels.

The prototype system was developed to simulate realistically the operation of the Spoornet system. A twelve pulse rectifier was built to power a 3.3 kV dc bus. A dc drive was developed that drives a 50 kW dc motor. This simulates the traction locomotive. An ac to ac converter was developed to drive a synchronous motor that was mounted on the same axis as the dc motor. This was used to simulate the traction load as well as differing track conditions such as up-hill, down-hill or flat track [T6].

A multilevel chopper was combined with a three-phase inverter to convert the 3.3 kV dc, first to 800 V dc, and then to 380 V three phase ac ([B3] to [B5]). This combination was placed in anti-parallel with the twelve pulse rectifier. The regenerative capabilities of the substations were also reproduced in the prototype system. Two methods of dumping the regenerated energy were implemented. The first is the method currently implemented in the traction substations. This is using large dc contactors that switches a resistive load according to the value of dc bus voltage. The second is a high voltage, IGBT composite switch that uses controlled pwm [T7] switching to limit the dc bus. This switch will be discussed in detail in chapter 4.

1.5 Operational Conditions

The prototype was designed to give the user the capability to study all typical conditions experienced in the Spoornet traction system. The power level differs from the system in the field, but the voltage levels are the same. The two extremes in the operation of both the practical Spoornet system as well as the prototype system is maximum powering mode and maximum regeneration mode. Figure 1-3 shows a simplified diagram of the entire system. The subsystems are shown by the blocks marked A through F.

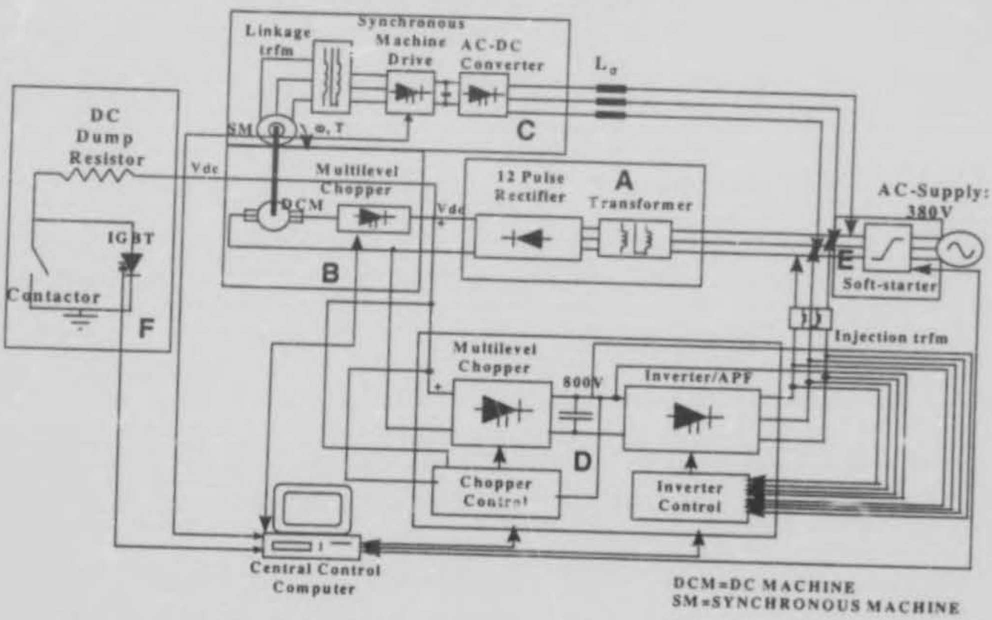


Figure 1-3 : Prototype system simplified block diagram

The author did not build the system alone, but were part of a team working on the prototype system. Table 1-3 summarises the manpower distribution in the construction of the system.

Table 1-3 : Manpower distribution in the construction of the prototype system

Subsystem	Person responsible	Section in Figure 1-3.
Twelve pulse rectifier	E. Beaud	A
DC motor drive + Control	C. Putter [B6]	B
AC Motor drive + Control	A.D. le Roux [B7]	C
DC to DC Converter	R.H. Wilkinson [B3]	D
Inverter	A. Horn [B3]	D
Soft-starter	E. Beaud	E
DC Dump + Control	E. Beaud	F
Overall system control	E. Beaud	

It can be seen that all the subsystems communicate with the central control computer. This is the point from which the user will be controlling the operation of the system to simulate the different operating conditions. In the running of the system all the system parameters can be studied and the effects of regeneration or powering studied.

1.5.1 Powering Mode

When the prototype system is operating in this mode it simulates power drawn by the traction locomotive. This would be when the train is moving up a gradient or if the train is busy accelerating or just normal cruising. In the prototype system as shown in Figure 1-3 the power will flow through the soft-starter (E) and twelve pulse rectifier (A) to the 3.3 kV dc bus. The dc motor drive (B) will accelerate the motor at the maximum acceleration. The synchronous motor drive (C) will be loading the motor by drawing power from the synchronous motor and injecting it back into the ac grid.

The only real power that will be flowing in the anti-parallel path (Block D) will be the losses incurred in the converter. The currents drawn by the rectifier will have a reactive as well as a harmonic component [A13], [A14]. The inverter will be injecting compensating currents into the ac grid to make the power factor of the substation unity and the current drawn sinusoidal. The algorithm ([A15] to [A17], [B8], [B9]) used for the inverter combines the charging of the 800 V dc bus with the injection of the compensating currents.

The central control computer is continually monitoring all the system parameters and checking for errors. The methodology of this will be discussed later. The control signals for the speed of the motor and the torque applied by the synchronous motor drive is generated by the central control computer. This implies that indirectly the power flow in the system is controlled by the user.

1.5.2 Regeneration Mode

Again the power flow in the prototype system is a scaled version of that experienced in the practical Spoornet system. In the practical system the maximum regeneration will be experienced when the train is braking against a down hill gradient or when decelerating. When the prototype system is in regeneration mode power will flow back through the dc motor drive (Block B). This power will be generated either by the deceleration of the rotor or by the synchronous motor drive (Block C) powering the motor. This will represent the deceleration of the train or down hill braking respectively.

The regenerated power cannot flow back through the rectifier because of its one directional nature. The dc bus will tend to rise in voltage because of the regenerated energy. The chopper in the anti-parallel path (Block D) will detect this rise in voltage and start drawing current from the 3.3 kV dc bus to the 800 V dc bus. The inverter will sense the consequent rise in the 800 V bus and inject real power back into the ac grid. The amount of power that can be regenerated is limited to about 200 kVA and any power that is regenerated above that limit will once again tend to have the dc bus rise. In that case the resistive dc dump (Block F) will sense the over-voltage and dump the energy into the large dc dump resistor. Once again two options are available as mentioned in section 1.4. The one being the dc contactor and the other the pwm controlled IGBT switch.

In the previous two sections the extremes of the operation of the prototype system were discussed. The operator of the system can control the system for any condition ranging from the one extreme to the other by controlling the speed and torque inputs to the respective motor drives. Some metering equipment has been installed measuring the dc bus voltages, the ac input voltage as well as currents showing the power flow throughout the system. The central control computer has also been fitted with an analogue to digital converter card that is able to read in different system parameters. This enables the user to monitor the operation of the system from a central point in all the modes of operation.

1.6 Report Structure

This first chapter has been introductory in describing the current practical dc traction system currently employed by the South African railway companies - Spoornet, Metro and SARCC. It also purposes to give a short introduction to the prototype system that has been developed in the laboratory to represent this system in a controlled environment. The following chapter will describe in more detail the development of the prototype system. With a good overview of the system the third chapter will focus on the protection strategies that have been employed in the current practical system. This will also be the starting point of the protection strategies discussed in this thesis.

The main protection device in the prototype system is a high voltage IGBT dc dump that has been developed. Chapter four will look in detail in the development and operation of this device using conventional control methods before chapter five investigates the operation of fuzzy logic in the control of the dc dump. The protection control theme is then expanded to the whole prototype system in chapter 6 where two "intelligent" controllers are examined. The operation of these controllers are discussed in chapter seven and some comparisons made. The thesis will end off with a summary of the work done in the thesis, some conclusions will be drawn concerning this work and a look will be taken at possible future continuations.

1.7 Summary

For this thesis a system has been developed that simulates the 3 kV dc traction system used in South Africa, Italy and Belgium. The prototype system is not an exact replica of the system in the field, but a scaled down version of it. The voltage levels are the same as in practice, but power levels are reduced by about 25 times. The thesis proposes additions to the current Spoornet system that enhances the regenerative capabilities of the traction substation as well as improves the harmonic content of the currents drawn by the substation. A central control strategy has been implemented that controls the operation of the prototype system as a whole.

Protection issues have also been addressed using the same controller that controls the system operation.

Chapter 2

Development of a 200 kVA Prototype System

2. Development of a 200 kVA Prototype System

2.1 Introduction

The purpose of building the prototype system was to have a laboratory set-up that represents the practical Spoornet traction system. The voltage levels of the prototype system are the same as that found in the practical system, but the power level is 25 times less. The voltage levels have to be similar because the main constraint on the switching elements is the high dc voltage. It is easier connecting the switching elements in parallel to increase the current capability of the system, than connecting the elements in series to increase the voltage capability. In order to keep the cost of the prototype system relatively low it was decided to keep the power rating low.

The first prototype that was developed by Peter-Jan Randewijk [B1], [B2] was rated at 10 kVA, with a dc voltage of about 200V. This system had only an inverter in the anti-parallel path. The IGBT switching elements could handle this voltage easily and single switching elements could be used. In the 200 kVA prototype system the voltage was increased to 3.3 kV dc as is found in the practical system. Because no IGBT switching elements are commonly available that can handle such a high dc voltage ([M6], [M7]), a solution was needed to accommodate the high dc voltage. Two solutions were considered. The one was using a multilevel inverter that connected IGBTs in series in each of the three phase arms used. The other option was using a multilevel chopper that converts the 3.3 kV dc to 800 V dc. This 800 V dc bus can then be inverted using a normal three phase inverter. The merits of the one versus the other is discussed in ([A18], [B3], [B10]. It was decided to use the multilevel chopper, inverter combination.

2.2 Rectifier

The rectifiers used in the practical Spoornet traction substations are standard twelve pulse rectifiers. The transformers used in the different substations were manufactured by different companies as well as the rectifiers. The diodes that were used also differ, as well as the winding of the transformers, but the voltage level and ripple frequency are the same for the different rectifier stations. The rectifier used in the prototype system is also a standard twelve pulse rectifier with the configuration shown in Figure 2-1.

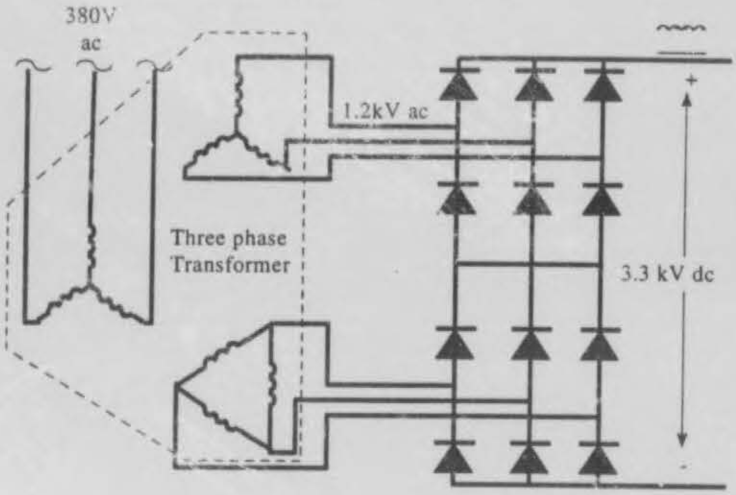


Figure 2-1: Twelve pulse rectifier

2.2.1 Transformer

The transformer has one primary winding and two secondary windings. The primary is a star wound three phase winding to eliminate circulating currents. Different taps were provided on the secondary to vary the output dc voltage for testing purposes. The taps were chosen for specific output voltages ranging from 1.8 kV to 4.5 kV. A range of possible output voltages were chosen to facilitate the change in voltage experienced on the overhead lines above the locomotive. It is also safer to work at a reduced voltage during the development phase.

The two secondaries were wound to have a phase difference of thirty degrees between the two. The secondaries feed two six pulse rectifiers connected in series. A six pulse rectifier produces a dc voltage with a 300 Hz ripple superimposed on it. The ripple repeats itself every 60°. If the one secondary is displaced 30° with respect to the other. The two ripple voltages will not interfere constructively with each other, but destructively. The frequency of the resultant ripple voltage will also double to 600 Hz.

The peak power rating of the transformer needs to be the same as the maximum rating of the forward path of power flow which is 200 kVA. The transformer is rated to provide a continuous 20 kVA source with a peak power rating of 200 kVA for 10 seconds. The specifications for the transformer is summarised in Table 2-1.

Table 2-1 : Transformer specifications

Taps	56 %	82 %	100 %	120 %	137 %
Input voltage	380 V				
Output voltage	1.8 kV	2.7 kV	3.3 kV	4 kV	4.5 kV
Continuous Power	20 kVA				
Peak Power	200 kVA for 10 seconds				
Impedance	3.2 %				

2.2.2 Rectifier Diodes

The diodes shown in Figure 2-1 actually consist of two series connected diodes. This was done to increase the voltage rating of the rectifier. The total voltage rating of the rectifier needs to be 4 500 V. Each six pulse rectifier needs to handle half of that voltage which is 2 250 V. When a diode in the bottom half of the bridge is conducting, the voltage across it is practically zero. This means that the voltage across the top diode is the full dc voltage of 2 250 V. With two diodes connected in series a voltage blocking capability of 1 125 V per diode is needed. The snubber circuit PCB is shown in Appendix A1.

The practical rectifier is shown in Figure 2-2. The current is measured by a resistive shunt connected in the earth return path.

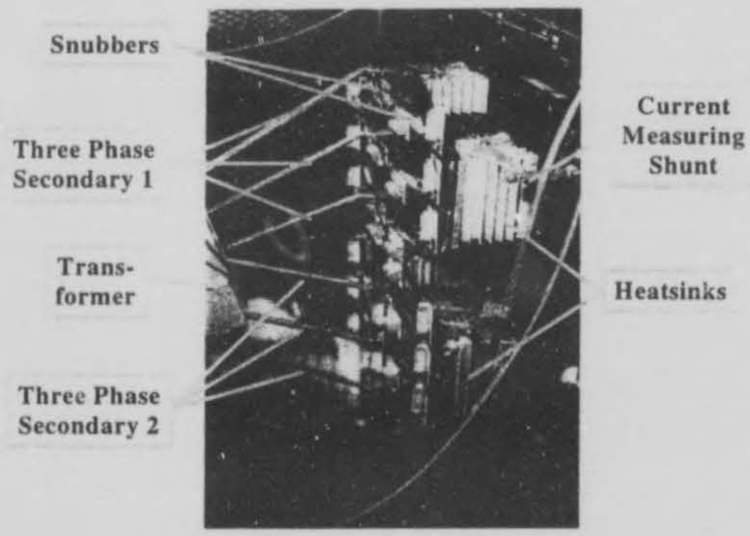


Figure 2-2 : Practical twelve pulse rectifier of the prototype system

2.3 DC Motor Drive - C. Putter [B6]

A two quadrant variable speed drive is needed for the dc machine to simulate the two modes of operation of a locomotive. These are the powering mode that simulates the normal path of power flow from the substation to the locomotive. The other is regeneration mode where the traction motors act as generators. The power then flows from the locomotive back to the substation. The drive for the dc machine have to be capable of handling the 4 kV dc overhead line voltage as well as the nominal and peak current in the machine. The topology used is shown in Figure 2-3.

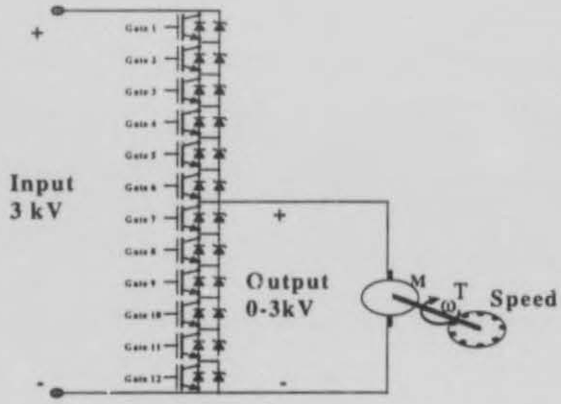


Figure 2-3 : High voltage dc motor drive [B6]

This drive performs speed control on the dc motor. The series connection of IGBTs need special protection, but more will be said about that in chapter 4. A photo of the practical converter is shown in Figure 2-4.

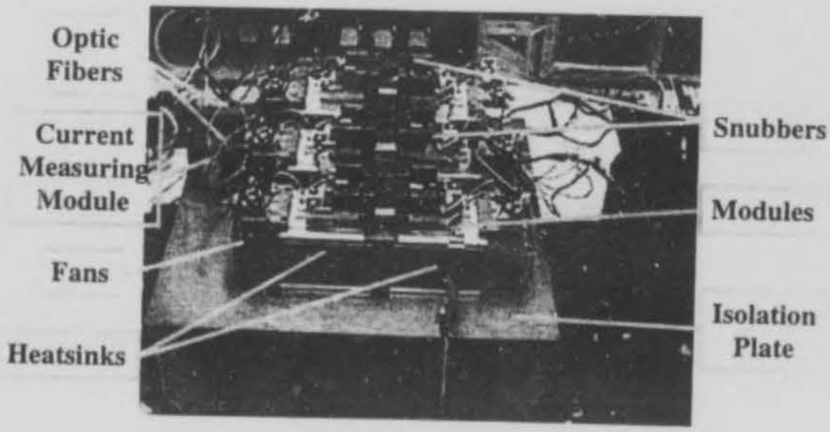


Figure 2-4 : High voltage dc motor drive

2.4 Synchronous Motor Drive - A.D. le Roux [B7]

The prototype system has a converter that simulates the locomotive of the train. It also needs a system to simulate a load effectively as the only mechanical load is the inertia and the friction of the rotor of the motor. This is accomplished through a synchronous motor that is mounted on the same axis as the dc motor. The synchronous motor is driven by a four quadrant ac to ac converter that is able to have the load act as a power sink as well as a power source. This would translate to up hill or down hill movement as well as prolonged breaking or acceleration.

The layout of the ac to ac converter is shown in Figure 2-5. The synchronous rectifier provides an 800 V dc bus that connects the rectifier to the inverter. The inverter uses vector control to drive the synchronous motor in the required mode of operation. The two converters can reverse roles as it is necessary for power to flow in any required direction.

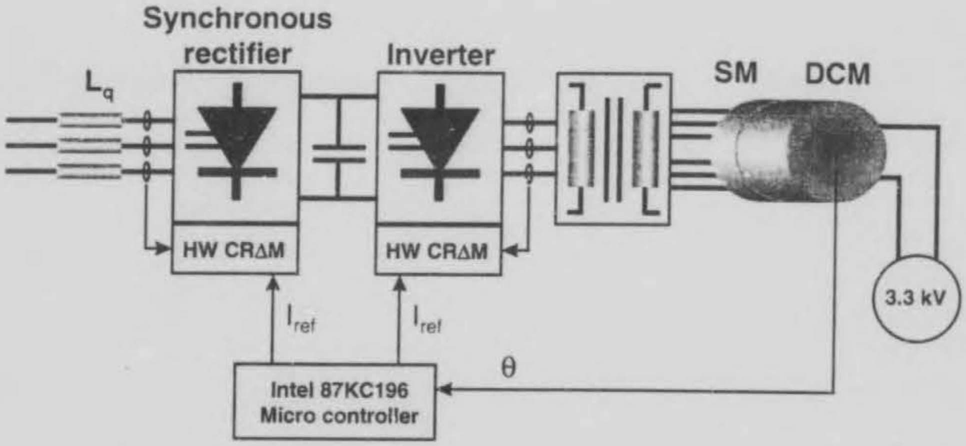


Figure 2-5 : Synchronous motor drive for load simulator [B7]

The synchronous motor drive is given a torque reference to implement on the synchronous motor. A positive torque applied would simulate a heavy traction load being pulled by the locomotive. A negative torque would be the same load pushing while the locomotive is braking. A photo of the practical ac to ac converter is shown in Figure 2-6.

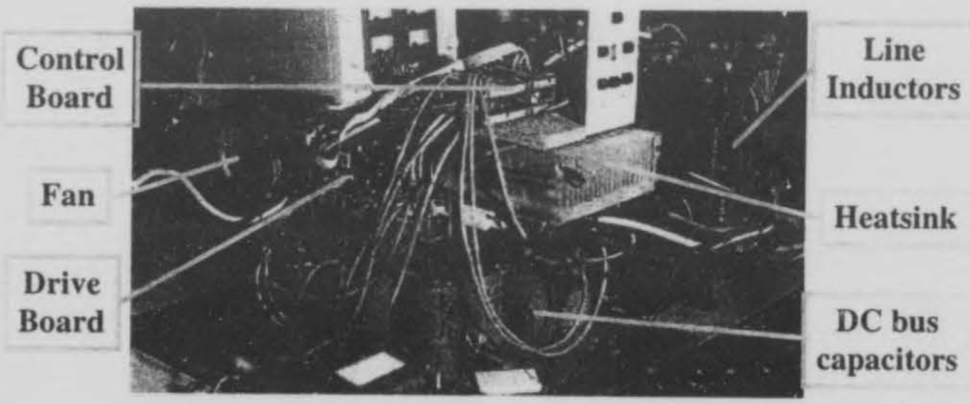


Figure 2-6 : Synchronous motor drive for traction load simulator

2.5 Anti-parallel Regeneration Chopper - R.H. Wilkinson

The injection of power, generated by the locomotive, back into the ac grid necessitates a path anti-parallel to the rectifier. Power cannot flow back through the rectifier so it either has to be dissipated, or it can flow to the ac side through an alternative path. If the latter alternative is chosen, the power needs to be injected into the ac grid in three phase ac form. This needs a three phase inverter to convert dc to three phase ac. The problem with this is the high dc voltage. Switching devices exist that can switch more than 3 kV, but the frequency of switching is very low. It was decided to use IGBT switches that switch fast, but also block a reasonable voltage.

The IGBT has a typical maximum blocking voltage of about 1 kV ([M6], [M7]). This means that a converter is needed to convert the 3 kV of the dc bus, down to about 800 V which is ideal for the dc side of the inverter. Several topologies were considered, but the normal buck converter was found to be the most cost efficient solution. The topology used is shown in Figure 2-7.

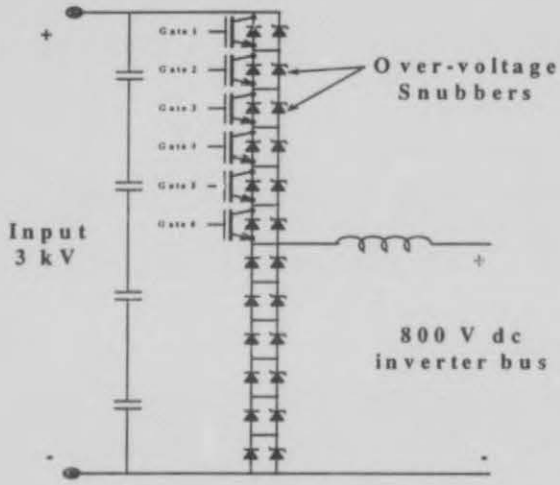


Figure 2-7 : Anti-parallel regeneration chopper

Both the switching devices as well as the free-wheel diodes need voltage limiting snubbers across every individual device to ensure correct voltage division. The power level of the converter is 200 kVA to match to the rest of the system. This is also the maximum amount of regenerative power that can be fed back through the anti-parallel path. During normal powering operation this converter will be idle. A photo of the practical converter is shown in Figure 2-8.

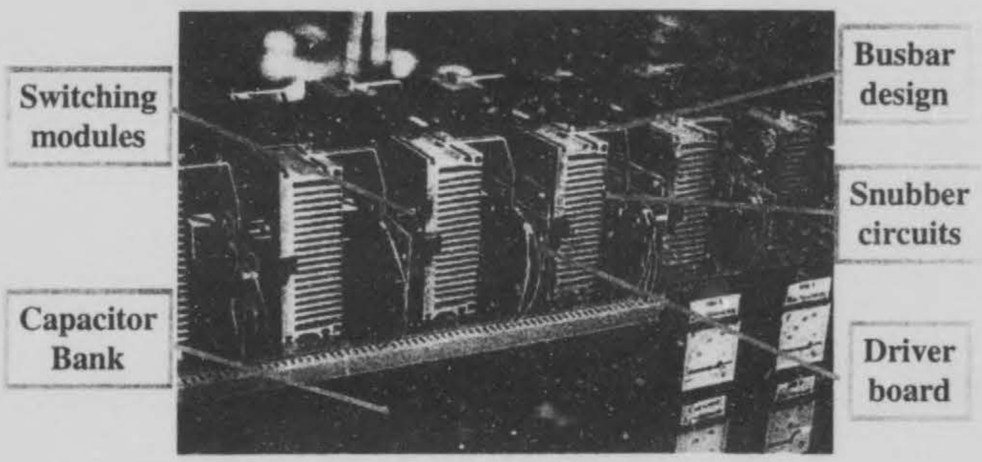


Figure 2-8 : Practical anti-parallel chopper [B6]

The converter is still in the testing phase and not yet connected to the rest of the system.

2.6 Inverter - A. Horn [T5]

This device is the main power quality improvement device in the system. The inverter acts as an active power filter during normal powering mode ([A19] to [A21], [B11], [B12]). The controller of the inverter calculates the difference between the ideal, unity power factor, sinusoidal waveform and the twelve pulse current waveform drawn by the rectifier. The inverter uses current control ([B13], [B14]) to inject the compensating current so that the substation looks like a resistive load to the ac grid.

The additional advantage of using this configuration in this application is that the inverter can inject sinusoidal currents back into the ac grid during regeneration. The configuration used is the standard three phase inverter shown in Figure 2-9. A power rating of 200 kVA was chosen to match that of the chopper.

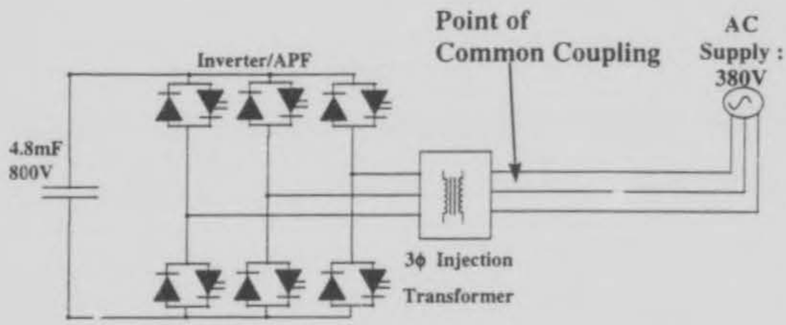


Figure 2-9 : Three phase active power filter inverter

The inverter was constructed implementing a busbar design that minimises the effects of EMI and also the amount of inductance between the dc bus and the switching elements. The control is done using a TMS320C31 DSP for the outer loop control and a TMS320C50 DSP for the inner loop [T8]. A photo of the converter is shown in Figure 2-10.

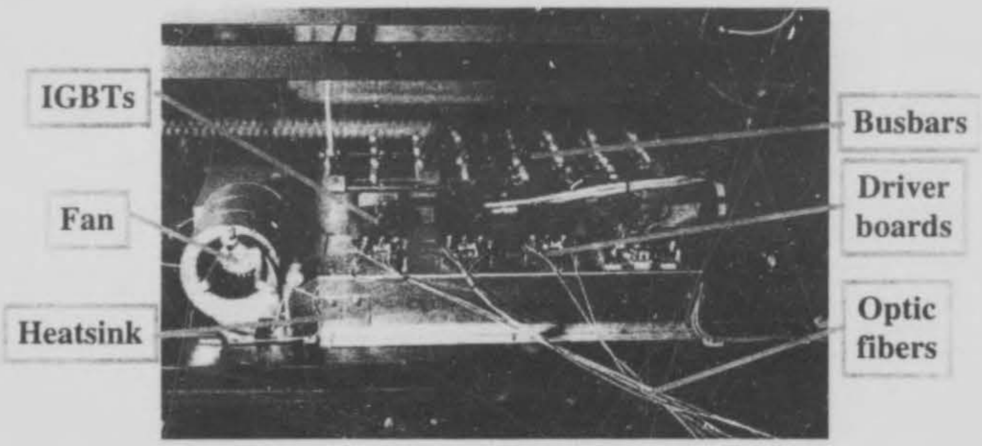


Figure 2-10 : Practical active power filter inverter [T5]

This converter has been operated apart from the rest of the system. It was used to do active power filtering on a six pulse harmonic load [T5], but it has not operated on a twelve pulse load nor done active power injection.

2.7 Soft-starter

In the operation of the prototype system the system will be switched on and off regularly. This is because of the research nature of the work being done. During switch-on of the system, all the dc buses need to be charged. A method of starting needs to be employed that limits the inrush currents during start-up. During fault conditions the system also needs to be isolated from the ac grid quickly to limit any damage that could occur. The device that was chosen for these purposes is a thyristor soft-starter as shown in Figure 2-11.

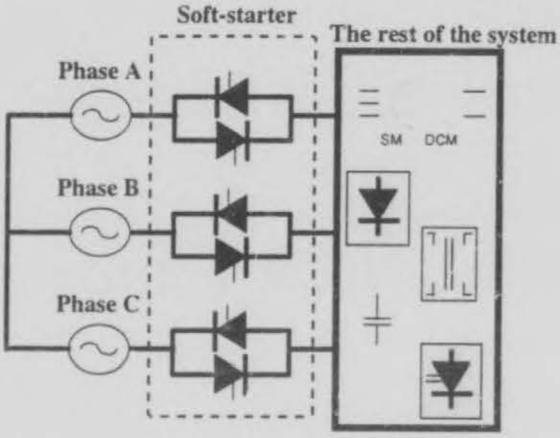


Figure 2-11: Three phase thyristor soft-starter

By controlling the firing angle of the thyristors the soft-starter can be used to regulate the main dc bus voltage. This was used extensively for testing purposes (Appendixes A2 to A4). A photo of the soft-starter is shown in Figure 2-12.

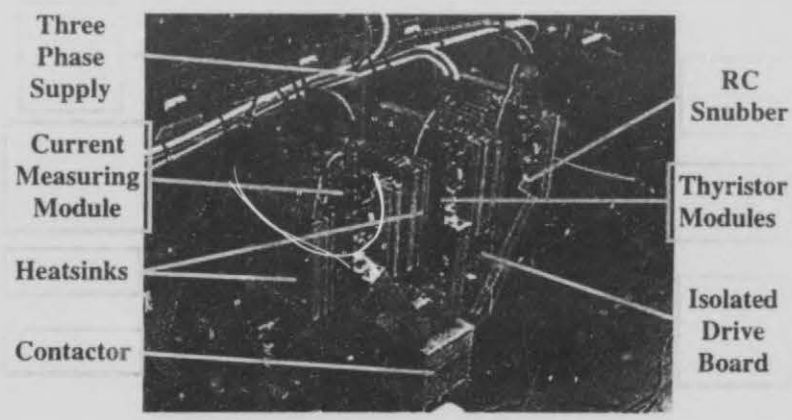


Figure 2-12 : Practical three phase soft-starter

2.8 Summary

A prototype system has been constructed containing all the essential components of the Spoornet dc traction system. This is the rectifier that generates the dc voltage on the overhead lines. The system also contains a dc drive that does speed control on a 3 kV dc motor. This motor is loaded by a synchronous motor drive that allows bi-directional power flow. This allows the system to operate in all the modes of operation encountered in the practical dc traction system. Excessive regenerated energy is dumped into a resistor bank via a 3 kV dc contactor as is also done in the practical Spoornet substations.

The proposed additions to the system is an anti-parallel path that allows regenerative power to be injected back into the ac grid. This is done by a high voltage chopper working in conjunction with a three phase inverter. The inverter is also used to do active power filtering during normal operation.

A photo of the entire system is shown in Figure 2-13.

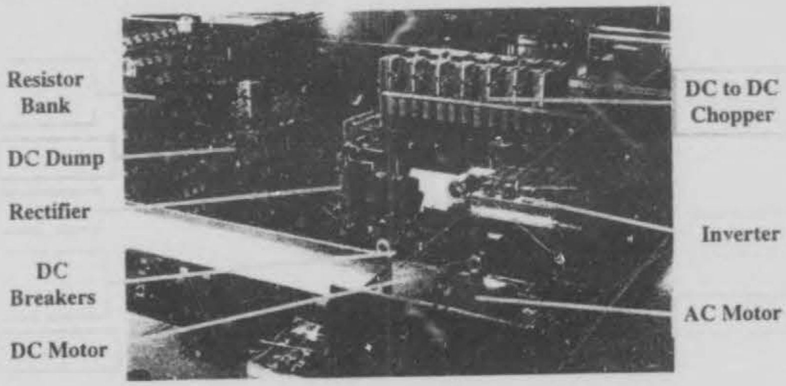


Figure 2-13 : Practical prototype dc traction substation

The operation of the prototype system makes it possible to do various system studies on the small scale system that would be equally applicable to the practical traction system. The effects of active power filtering and many other studies can be done that is indicative of the practical dc traction system.

Chapter 3

Current Substation Protection Strategies

3. Current Substation Protection Strategies

3.1 Introduction

In most of the power systems currently employed in the field a protection network is installed that protects the system against diverse fault conditions. The protection devices are mostly dedicated for a specific purpose and designed accordingly. By the correct combination of all the devices, the system is protected against all the commonly encountered fault conditions. In this chapter the implementation of this classical protection method is discussed in the context of the Spoornet dc traction substation [M1].

In the typical Spoornet substation the ac power is supplied via a power switching yard situated next to the traction substation building. This is an outdoor facility feeding the transformer which is usually standing outside. The rest of the substation which includes the rectifier, the filter coil and other filtering and protection equipment is located inside the substation building. The protection strategy can be similarly divided into an ac and a dc side protection scheme.

3.2 AC Side Protection Levels

On the ac side there are three levels of protection each having its own purpose. The ac side does not have problems with breaking dc current, but the voltage levels on the ac side is much higher (From 11 kV to 132 kV). ESCOM is the national utility and it has installed a motor operated disconnector between the ESCOM network and the Spoornet side of the grid. This is to isolate the ESCOM grid from the Spoornet side during maintenance.

The ac side of a typical Spoornet substation is shown in Figure 3-1. It can be seen that an auxiliary transformer is used to provide standard 230 V ac power to the rest of the substation. The power rating of the auxiliary power system is usually 50 kVA with a relatively small protection relay for fault conditions.

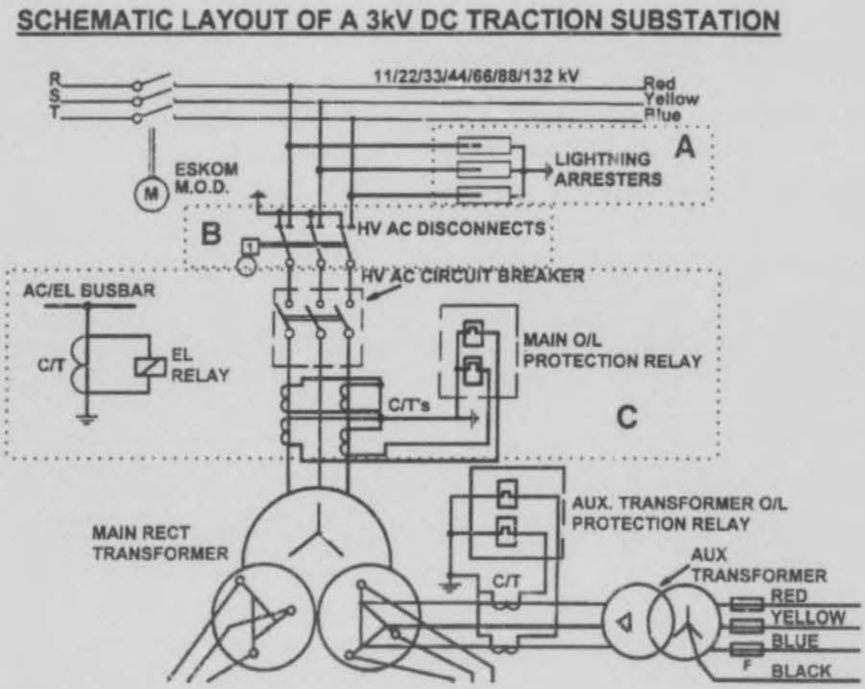


Figure 3-1 : AC side protection in a 3 kV dc traction substation [M1]

3.2.1 Lightning Arresters

Because the ac side of the substation is outside in the yard it is highly likely that it will be hit by lightning. Normal grounding practices are followed, but still the lines has to be protected in the case of a direct hit. The first layer of protection is the lightning arresters that monitor the lines for over-voltage. The arresters are placed in parallel with the rest of the substation as shown in Section A of Figure 3-1. In the case of a hit the voltage on the line will rise and the lightning arrester will flash over. The charge of the lightning flash will flow down the lightning arrester to ground and the rest of the substation will not be adversely affected.

3.2.2 High Voltage maintenance disconnectors

These are mechanical contacts that are opened to isolate the substation from the rest of the ac grid. This is done mainly for maintenance purposes. These contacts are not designed to open under load and therefore they are mechanically and electrically interlocked with the main circuit breakers. They cannot open unless the main circuit breakers are open and there is no current flowing in the conductors. These disconnects are the first protection device placed in series with the rest of the substation and are shown in section B of Figure 3-1.

3.2.3 Main Circuit Breakers

These devices are the main protection unit on the ac side of the substation. These circuit breakers are designed to break up to 30 kA of current during fault conditions. These circuit breakers are controlled electrically and mechanically. The electrical control circuit monitor the earth leakage current as well as the current drawn by the substation. In the case of overloading or earth leakage the circuit breaker is opened, isolating the substation from the ac grid. The circuit breakers as well as the controlling circuitry is shown in section C of Figure 3-1.

3.3 DC Side Protection Levels

The transformer feeding the rectifier stands outside the substation building. High voltage, high current feeders go into the building to feed the rectifier that produce a 3 kV dc voltage. Protection of the dc side of the rectifier is very different to the ac side because of the lack of zero crossing of the current. This makes any circuit breaking a lot more difficult. Another challenge on the dc side is the harmonics generated by the rectifier. This differs from a fundamental ripple voltage frequency of 600 Hz with the twelve pulse rectifiers, to a fundamental frequency of 300 Hz with the six pulse rectifiers. Figure 3-2 shows the dc side of a typical traction substation. The different subsections of the sketch show the protection circuitry and they will be discussed shortly. It can be seen that the main strategy of the protection scheme is dealing with harmonics and containing the influence of a substation to the section that it is feeding.

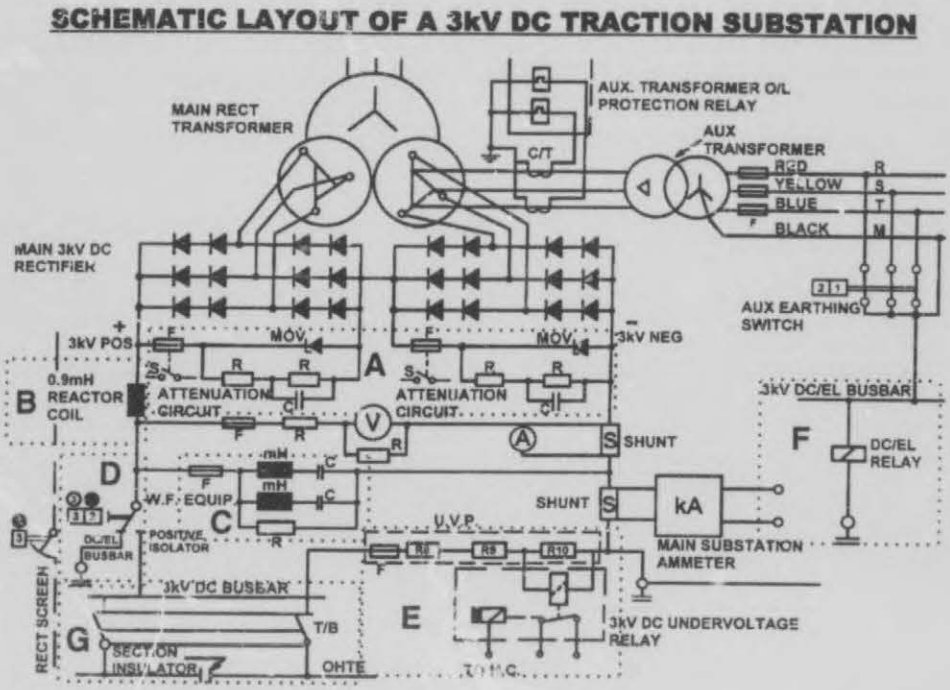


Figure 3-2 : DC side protection in a 3 kV dc traction substation [M1]

3.3.1 Over-voltage MOV's

The first layer of protection after the rectifier is the over-voltage protection. This is done by a 4.8 kV, 5 000 A (peak) zener diode. The diode is protected by a fuse (6 kV, 20 A) in series with the device so that the protection device cannot draw too much current. It is therefore a current protection of an over-voltage protection device. The over-voltage protection is done separately for each six pulse rectifier with totally independent attenuation circuits. These circuits with their RC snubber circuits are shown in section A of Figure 3-2.

3.3.2 Reactor Coil and Waveform Equipment

These are not protection devices in the classical sense of the word in that they do not protect the system against a fault condition. They do protect the system against the effects of the harmonics induced by the rectifier. These ac voltages on the dc bus transmit and influence the signalling gear next to the dc tracks. This leads to false signalling that could lead to major disasters. They also cause corona effects and heating losses in the dc equipment. The coil in the substations is 0.9 mH and is shown in section B in Figure 3-2 and the filtering equipment in section C. The inductor also limits the dI/dt during fault conditions which makes it possible to set the fault levels lower on the breakers. The filtering equipment is tuned to either the 12th and 24th harmonics for 12 pulse rectifiers or for the 6th, 12th, 18th, and 24th harmonics for six pulse rectifiers.

3.3.3 Positive Isolator

The positive isolator is installed between the rectifier and the positive 3 kV dc busbar for the purpose of isolating and earthing the rectifier output during maintenance. These isolators are provided with mechanical interlocks to prevent the switch being operated unless the primary circuit [illegible] and the HV ac disconnecting switch have been opened. This switch is not designed as a circuit breaker and can not be opened under load.

3.3.4 Earth Leakage Relay

There are three separate earth leakage monitor circuits that combine to protect the substation. These monitor the ac side, the dc side as well as the auxiliary power supplies for earth leakage faults. These will detect any flash-over to ground, insulation failure or accidental shock caused by staff.

3.3.5 Under-voltage Relay

This consists of a resistive divider operating as a voltage monitor. This drives a relay that opens the track breaker holding coil if the bus voltage goes too low. This could happen under various conditions like overloading or a faulty supply. Independent of what the fault condition is, the substation is isolated from the overhead tracks so that operation of the rest of the system can continue.

3.3.6 Section Isolator and Track Breaker

A track breaker is a high speed dc circuit breaker. Two of these are used between the 3 kV busbar and the individual overhead track equipment circuits to protect the overhead conductors against overloading and high fault currents. The breakers are rated to interrupts dc currents in the order of 10 kA and have extensive protection against the inevitable arcing that occur during interruption.

Each substation feeds two sections of tracks. The tracks are separated into sections of between 8 km and 20 km by section isolators at the substations. This means that there are always two substations feeding any one section of track giving greater system reliability.

3.4 Conclusions

The above strategy is a very thorough protection scheme that protects the system against all typical error conditions. It can be noted that the devices used are all local protection devices, meaning that each monitor a specific parameter, like the bus voltage, and switch a protection device according to what could have caused the error as measured.

This is very reliable, but it could be argued oversensitive. It does not matter what the fault condition, the whole substation will be out of operation in the case of an error. If this is not totally necessary, the system could be given a ride-through capability or intelligent system response to avert the downtime caused by an unnecessary substation shutdown.

Chapter 4

A 3 kV Solid State DC Brake

4. A 3 kV Solid State DC Brake

4.1 Introduction

In the practical set-up the power rating of the anti-parallel regenerative path of energy flow would be in the order of 400-600 kVA in the final design. Yet the amount of energy regenerated at a steep down-hill gradient is up to 2.7 MVA [M1]. This means that a backup method of energy dissipation is necessary to limit the voltage rise on the overhead lines due to regeneration. In the current practical system 100 A dc contactors (Appendix A5) are used to switch 33 Ω resistor banks onto the overhead lines to dump the regenerated energy. Up to 9 banks (100 A per bank) are provided that switch in stages to give the substation continuous regenerative capabilities (Maximum 900 A). Because of the resistance of the overhead conductors, the distance over which regenerative operation is possible is limited to between 112 km (100 A) to 3.6 km (900 A) [M1]. The locomotive stops regeneration at an overhead dc voltage of 3 900 V.

Several problems were encountered with the dc contactors.

- Because of continuous arcing in the breakers the contacts needed to be replaced regularly.
- The fast varying voltage during regeneration required the switches to switch very regularly. This increased wear on the switches.
- The switches operate at a very low frequency giving a measure of discrete operation to the regenerative operation. A smoother operation would be desirable.
- The switches operate at a very unique voltage (3.3 kV dc range) making general availability low and replacement very difficult.

A silicon alternative was suggested to replace the dc circuit breakers. This solution addressed all the above mentioned problems while performing the task of regenerating excess energy equally effectively. This circuit for this solution is show in Figure 4-1.

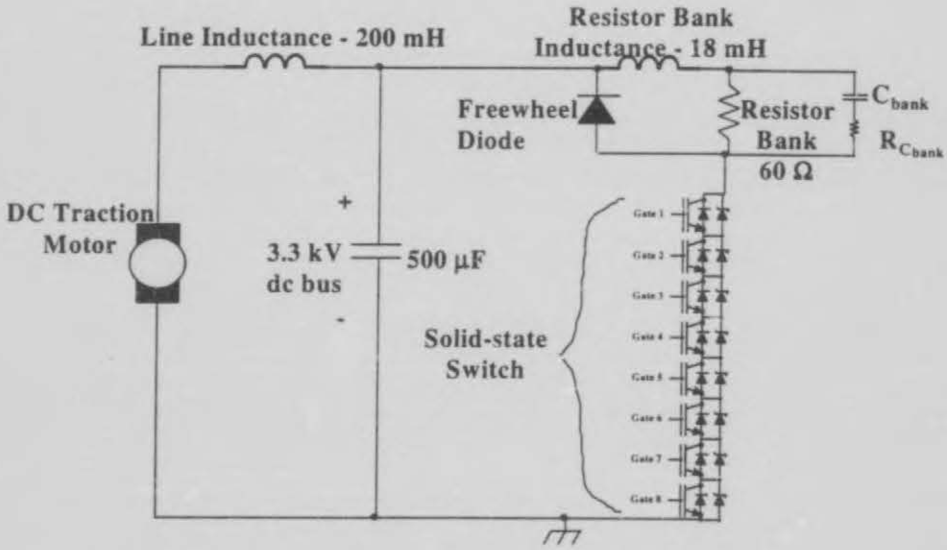


Figure 4-1 : **Solid state dc dump switch for regenerative power dissipation**

4.2 Development of the solid state dc dump

4.2.1 Introduction

In the practical system the dc contactor switches a resistor bank onto the overhead lines to dump regenerated energy. The solid state dc dump does exactly the same. The main difference is that the one does it mechanically while the other does it using semiconductor technology. This enables faster switching without the arcing associated with mechanical dc switching. The semiconductor switch needs auxiliary components to facilitate switching without over-voltage and this requires higher costs, but the lack of moving parts brings down maintenance costs considerably.

It was decided to develop the first prototype of the dc dump switch as a scaled down version of the final product. This was done to reduce costs in the development stage. Power levels were chosen to be representative of that in the final installation yet lower by a factor of 2-3. The voltage levels were not reduced as this is the main constraint in the technology that is being applied. A series connection of devices were chosen to overcome the problem of the high dc voltage. Two control methods were evaluated being PI control and fuzzy control.

4.2.2 Switching elements

The rating that was agreed for the switch was 50 A continuous, 4000 V dc. This does not elaborate on the speed of switching required nor has any reference been made to the driving circuitry or the control. These criteria were decided upon by the author.

The current specification indicates that 50 A needs to be the maximum continuous current flowing through the switch. Semiconductor switching devices exist that have a 50 A continuous rating, but it is extremely difficult keeping these devices cool. Elaborate cooling techniques have to be implemented to operate these devices at this rating. It was decided rather to use devices with a higher current rating, but with better thermal characteristics.

The voltage rating needs to be considered keeping the type of device used in mind. This has not been determined yet, so the voltage rating will be discussed later.

The type of switching device used will be largely determined by the switching speed required and the current rating. One of the objectives in the improved switch was to have a smoother operation due to a higher switching frequency. It was also thought to research high frequency, high voltage operation. The faster switching elements currently available is the mosfet and the IGBT. The mosfet can switch considerably faster than the IGBT, but the current ratings available is not sufficient for this application. It was decided to use a 200 A IGBT because of the good thermal characteristics. It was also available in modules of two IGBTs per module which makes construction a lot simpler. The highest available voltage rating was 1200 V per IGBT. Keeping a safety margin of 100 % it was necessary to use 4 modules i.e. 8 IGBTs in series to ensure a sufficient voltage capability.

4.2.2.1 Heatsinks

To calculate the heatsink necessary for keeping the IGBTs cool the total power loss in the IGBTs has to be calculated first. The freewheel diodes in the IGBT modules are not being used in this converter so all the heat that needs to be dissipated is generated by the IGBTs. The calculations for the powerloss in the IGBTs are done in detail in [M8], but a short summary will be given here.

The one main area of power loss is that incurred during conduction. The typical on-state voltage of the IGBTs is 3V and with a current of 50 A this implies a maximum loss of 150 W. This will occur at unity duty cycle. The other main loss element is the switching losses. This is the peak power dissipation during a switching transient where the voltage across the device as well as the current through the device is simultaneously high. The total switching losses is dependant on the frequency and therefore the switching frequency is limited by thermal considerations. A maximum allowed switching frequency of 2 kHz was chosen which necessitated total switching losses of 54 W. The total power loss per IGBT is then 204 W. This is not the final chosen switching frequency, but rather an upper limit to the choice of switching frequency.

The thermal circuit for one IGBT module containing two IGBTs is shown in Figure 4-2 [M8].

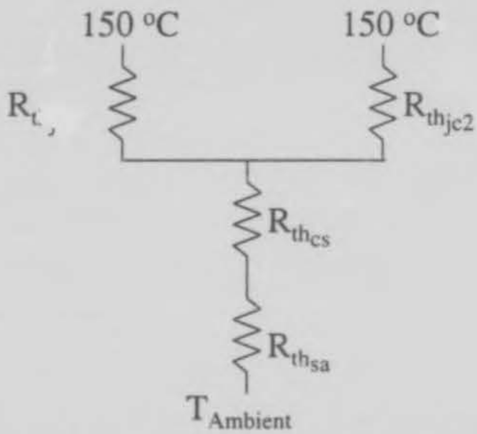


Figure 4-2 : Thermal Circuit Configuration [M8]

The junction to case thermal resistance (R_{thjc1}) is given in the data sheets as 0.1 °C/W. The heatsink is thermally bonded to the heatsink using thermal paste and the thermal resistance (R_{thcs}) is estimated to be 0.038 °C/W. With the total power dissipation being 2 x 204 W = 408 W and the ambient taken as 25 °C the thermal resistance of the heatsink (R_{thsa}) can be calculated to be 0.218 °C/W. A heatsink was available with a thermal resistance of 0.145 °C/W when combined with a fan. Every heatsink was electrically connected to the centre point of each module to protect the module against over-voltage between the IGBTs and the base plate. This necessitated the isolation of the base plates from each other and from ground.

4.2.2.2 Drive boards

The gate of the IGBT is driven similar to that of a mosfet. It looks to the driving circuitry like a capacitor that needs to be charged and discharged. A chip (CPA 7667 by Harris Semiconductor) is available that is designed to drive a mosfet gate. The chip is designed to inject a large peak current during switch-on and draw a large peak current during switch-off. The capacitance of the IGBT gate needs peak currents to switch fast and therefore this chip is ideally suited for this application. It works just as effectively in driving an IGBT gate. The main problems that needed to be overcome in the driving of the series connection of IGBTs was the isolation of the driving circuits to be floating with the emitter of each IGBT. Another requirement was that the IGBTs have to be driven simultaneously.

The first problem of the isolation was addressed by using a toroidal isolated power supply as shown in Figure 4-3. The toroids were designed to operate at high frequency and to provide about 10 kV of isolation between the primary and the secondary (Appendix A6).

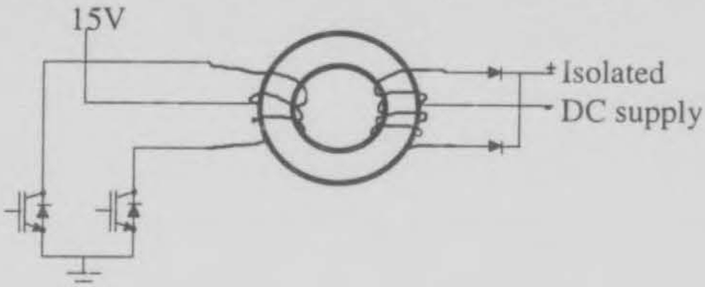


Figure 4-3 : Isolated dc power supply for IGBT gate drivers

The isolated dc supply is smoothed with capacitors and regulated to give a constant 15 V as well as a 5 V supply. The detail description and the full circuit diagram of the isolated power supplies is shown in [M8].

To have the IGBTs switching simultaneously the driving signal from the control board has to be the same for all the IGBTs. The driving circuitry of the individual IGBTs also have to react very fast to have the driving signal at the IGBTs at the same time. For this reason it was decided to use optic fibre interfaces with a capability of 5 Mbit/sec to convey the driving signal to the IGBTs. The advantage of this lies dually in the inherent isolation of the signal as well as the speed of response of the sending/receiving combination. The total isolated driving circuit for the IGBTs is shown in Appendix A2 of [M8] and Appendix A7.

4.2.2.3 Snubber circuits

In the series connection of IGBTs the problem is having the voltage share sufficiently between the IGBTs so that none of the IGBTs sees a voltage higher than it is rated for. There exists several ways of accomplishing this ([B3] to [B5]). Some criteria for choosing a method is the speed of response of the sharing device, the power dissipated by the sharing device and effect the device has on the switching waveform. The solution that was chosen to accomplish sharing between the IGBTs was a variation of the classical over-voltage snubber. The new snubber that was developed by C. Putter [T9] is shown in Figure 4-4.

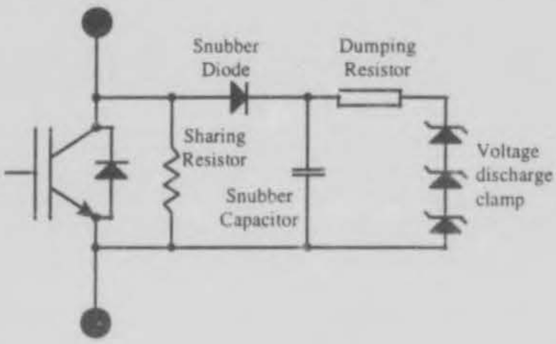


Figure 4-4 : Over-voltage snubber for series connected IGBTs

The snubber does not force sharing during switching, but works on the principle of operation when necessary. When the string of IGBTs switch off the voltage across the string rises to 3 kV (The nominal bus voltage). As the voltage across the switch rises for the first time, the snubber capacitors are charged up to some dc voltage depending on the natural sharing of the IGBTs. If the voltage rises above 800V (The limit of the voltage discharge clamp) the clamp starts conducting thereby discharging the capacitor through the dumping resistor. When the IGBT switches on again the diode is reverse biased and the capacitor maintains its voltage at 800 V. The sharing during switch-off depends mainly on the order in which the IGBTs switch off and the tolerances in the snubber capacitors. The sharing resistors function in conjunction with the snubbers to have the voltage share equally over the IGBTs over a long time. The current flowing in the sharing resistors is not comparable with that flowing in the snubber capacitors. This results in the voltage not sharing equally during the switch-off transient.

4.2.3 Capacitor Bank

During regeneration the current flows from the dc traction motor, through the overhead lines, through the dc contactor to the resistor banks. When the contactor opens the inductance in the path will tend to have the dc bus rise. If the contactor is to be replaced with a solid state switch, the rise in the dc voltage has to be contained to prevent the switch from being destroyed. The energy stored in the line and motor inductance will be stored in the dc bus capacitor (Figure 4-1) to be used again in powering mode. To calculate the energy stored in the path inductance the worst case line inductance has to be known. Because of difficulties in obtaining the required inductance from Spoornet, the inductance had to be roughly calculated. From [T13] the estimated equation for the line inductance per meter can be written.

$L_{line/meter} \quad = 4 \times 10^{-7} \ln (D/r')$

Where D is the distance between the lines and r' is the effective radius of the line. The overhead lines are estimated at 5 m constant above the ground and the ground is estimated as a single conductor at ground level. The line diameter was taken as 10 mm.

$r' \quad = 0.7788 \times 5 \times 10^{-3}$

$\quad = 3.894 \times 10^{-3}$ m

With D = 5 m this give an inductance per meter of

$L_{line/meter} \quad = 4 \times 10^{-7} \ln (5 / 3.894 \times 10^{-3})$

$\quad = 2.863\ \mu H/m$

With the maximum distance of line between the train and the substation taken as 45 km [M9] the total maximum line inductance will be

$L_{line\ total} \quad = 2.863\ \mu H/m \times 45\ 000$ m

$\quad = 129$ mH

Because of the estimations made a large safety factor has to be built into the design. The line inductance plus motor inductance was taken to be 200 mH. A reference [M9] that was found later gave a measured inductance of 0.970 mH/km. This is in the same order as the calculated results, but smaller. This means that the worst case assumption for the inductance was greater than the actual inductance. The resultant capacitor bank would therefore be more than adequate to clamp the rise in voltage.

With a maximum current of 50 A (The rating of the switch) being drawn by the switch the energy in the line will be

$E_{line} = 0.5 \times L \times I^2$

$= 0.5 \times 200m \times 50^2$

$= 250$ J

As this energy is stored in the dc bus capacitor, the voltage across the capacitor rises. A maximum rise of voltage of 1000 V was chosen to protect the components of the dc switch.

$E_{cap} = E_{line} = 0.5 \times C \times \Delta V^2$

$C = E_{cap} / (0.5 \times \Delta V^2)$

$= 250 \text{ J} / (0.5 \times 1000 \text{ V} \times 1000 \text{ V})$

$= 500\ \mu F$

With a maximum operating voltage of 4 000 V and a rated rise of voltage of 1000 V a total voltage rating of 5 000V is required of the capacitor. The dc bus capacitor had to be composed of 24 smaller capacitors with ratings 3300 μF , 450 V. 33 kΩ, 10 Watt voltage sharing resistors were placed in parallel with the capacitors to ensure correct voltage sharing. The connection of the capacitors is shown in Appendix A4 of [M8].

4.2.4 Resistor bank

The standard resistor banks used by Spoornet for regenerative braking is rated at 100 A and 33 Ω. This is to match the current and voltage rating of the dc contactor that switches the bank onto the dc bus. The current rating of the dc dump switch is 50 A and therefore a higher resistance is needed per resistors bank. Two standard resistor banks in series will give a total of 66 Ω which will limit the current to 50 A. With resistors donated by Spoornet a resistor bank was constructed using resistors rated at 100 A. These differed in resistance from about 0.2 Ω to 0.5 Ω and a total of 170 individual resistors was necessary to build a 60 Ω composite resistor (Refer to Figure 4-1). The inductive part of the load resistor will be discussed later. The load resistor also has a parasitic capacitance with its series resistance. The effect of this capacitance is a lot smaller than that of the inductance and was therefore ignored.

4.2.5 Freewheel diode

The above mentioned resistors are not purely resistive. They are wound resistors which gives them a significant inductive component. In the switching of this resistor/inductor between the dc bus and ground (Refer to Figure 4-1) the switch on is no problem. The current will rise gradually to its final value in a time proportional to the L/R time constant of the resistor. During switch off a problem arises. Because of the inductance of the resistor the current cannot be interrupted immediately. Because of the fast switching time of the IGBTs an alternative route for current flow has to be provided. For this reason a diode is placed in anti-parallel with the resistor so that the current can commutate between the switch and the diode during switch off. The energy stored in the inductance of the resistor will then be dissipated in the resistor itself as the current freewheels through the diode.

4.2.5.1 Diodes

The voltage rating of the freewheel diode needs to be the same as that of the switch. When the switch is conducting the diode is blocking the dc bus voltage and when the diode is conducting the switch is blocking the voltage. Because of the high inductance of the resistor bank the current rating of the diode needs to be in the same order as that of the switch. It can be reduced by a factor of two because the worst case duty cycle for the freewheel diode is 50 %. The steady state requirements of the freewheel diode is then 5000 V and 100 A. It is also required that the switching time of the diode be in the same order as that of the IGBT switch. The switch operates in about 1 μs so a faster switching time is needed from the diodes. Fast switching diodes manufactured by Semikron (SKD140F17) [M6] were used with a rating of 1700 V blocking voltage and 140 A average current. Three diodes were connected in series to give the required voltage rating. The rated switching times of these devices was 800 ns maximum.

4.2.5.2 Sharing resistors

The diodes used are also sensitive to over-voltage so it is necessary to ensure correct voltage sharing across the devices. This is accomplished by again using sharing resistors. A sharing resistor of 15 W, 150 kΩ was placed across every diode which at 1700 V gives a sharing current of 11 mA. This is a lot larger than the leakage current of the diodes and voltage sharing is thereby ensured.

4.2.5.3 Snubber circuits

When the switch switches on and the current is still flowing in the freewheel diode, the current has to commutate from the diode to the switch. The way the current switches off in the diode is shown in Figure 4-5. It can be seen how the current reduces gradually due to the stray inductance in the diode path. It does however not stop at zero current but goes negative in order to reverse bias the diode. When the PN junction is fully reverse biased the current snaps off to zero. The inductance that caused the gradual fall of current through the diode also conducts the current in the negative direction. This implies a theoretical voltage across the inductance of negative infinity to stop the current through the diode immediately. This means that an RC circuit is needed across each diode to limit the voltage across the diode during current snap off.

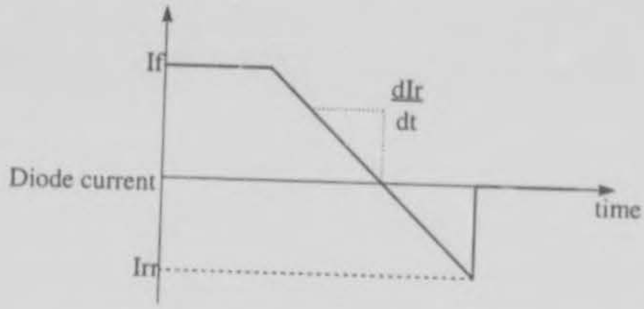


Figure 4-5 : Diode current during turn off

The design methodology is discussed in Appendix B and only the results for this design is shown here. The resistor rating is 10 Ω, 10 W and the capacitor is 3.2 nF, 2500 V. These are placed in series with each other and the combination in parallel with each diode.

4.2.6 Voltage measurement

The dc dump is a voltage controlled device. It monitors the voltage on its dc bus and reacts if the voltage goes too high. For this reason the voltage of the dc bus has to be converted to a value that the control circuitry can handle. The maximum voltage of the dc bus is 5000 V and the maximum voltage of the control circuitry is 5 V. Therefore a ratio is needed of 1000 between the measured signal and the real dc bus voltage. The measuring circuit is shown in Figure 4-6.

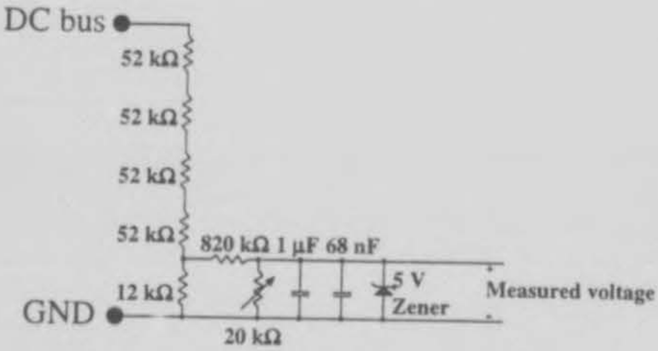


Figure 4-6 : Voltage measuring circuit for the 3 kV dc bus

For the purpose of minimising the effect of noise the voltage division was done in stages. High power, high voltage measuring resistors was used to divide the voltage down to 272 V maximum. This was then further divided and filtered down to 5 V maximum with the ration set by the value of the potentiometer. The filter capacitors were chosen to give a time constant of about 20 ms to be fast enough to control the dc bus, but slow enough to limit high frequency noise. A large capacitor was used to provide the required capacitance for the filter and a smaller capacitor was put in parallel to provide a low impedance path for high frequency signals. To protect the electronic circuitry a 5 V zener diode was placed across the measured signal to prevent the measured signal from going over 5V.

4.3 Control strategies for the solid state dc dump

The solid state dc dump is the main protection device in the prototype dc traction system. It measures the voltage on the dc bus of the system and, in the case of over-voltage, pulls it down by dissipating energy in a large dump resistor. Two control methods were implemented in operating the dc dump. The one was classical proportional/integral (PI) control and the other an implementation of a fuzzy logic control algorithm. Both methods used duty cycle control of the PWM switching waveform to control the dc bus voltage. The mathematical model of the calculation of the duty cycle differs dramatically between the two methods.

4.3.1 PI Control

The dc dump has to be pre-set to a value where it starts to operate. In other words it needs a reference level at which it will start having a non-zero duty cycle. The control loop for the PI controller is shown in Figure 4-7.

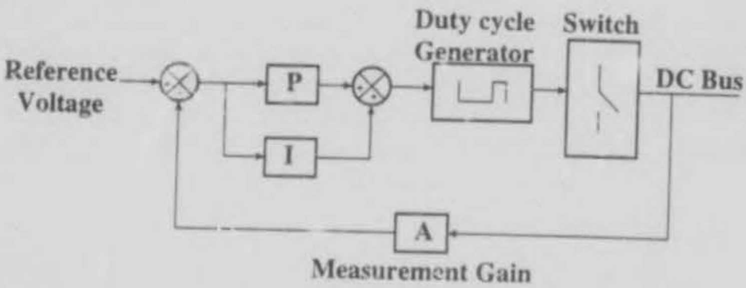


Figure 4-7 : Control loop for the dc dump using PI control

The switching of the 3 kV voltage causes large voltage transients causing much noise. The control circuitry had to be designed to be as immune to noise as possible. Smoothing capacitors were used where possible and all long wiring were shielded. The gain of the measurement (A) is 1/1000 as discussed in section 4.2.6. The design of the PI controller is shown in Appendix C. The proportional gain (P) is unity and the integral gain (I) is 1670. The circuit diagrams for the control boards as well as the board layouts are shown in detail in respectively Appendix A5 and B4 in [M8].

4.3.2 Fuzzy Control

The other method that was employed in the control of the dc dump was fuzzy logic control. This used a PC with an analogue/digital interface card to control the dump. The same measurement system was used as for the PI controller and the same duty cycle generator, but the control was done digitally by calculation in the PC. The theory employed in the application of fuzzy logic has not been discussed yet, but will be done in the following chapter. The operation of the dc dump under fuzzy control will also be investigated further in the next chapter.

4.4 Practical results of the solid state dc dump

The converter was built up according to the design as discussed. A photo of the practical converter is shown in Figure 4-8.

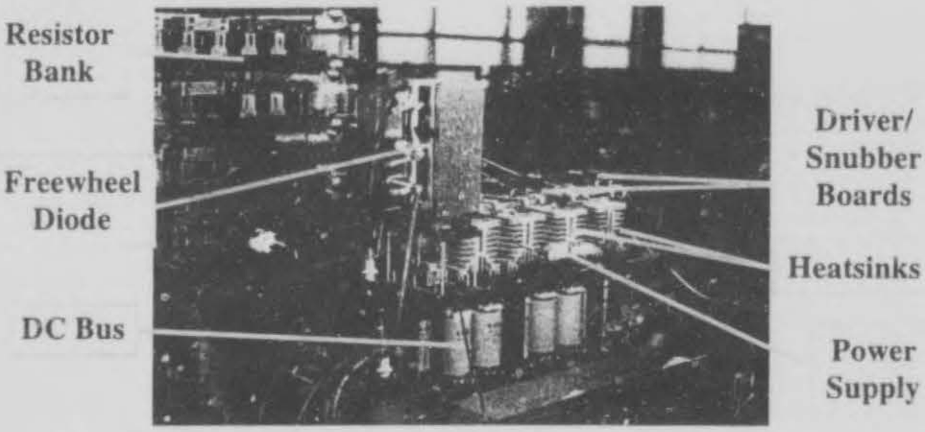


Figure 4-8 : Practical solid state dc dump switch

In the study of the operation of the dc dump two time scales are of importance. The first is the voltage and current waveforms of the IGBTs during switching. The time scale for this is in the order of microseconds, similar to the switching time of a single IGBT. The operation of the PI controller is in the time scale of seconds. This is in the scale of milliseconds as the controller uses variation in the duty cycle of the 1 kHz switching waveform to control the dc bus voltage.

4.4.1 Switching waveforms

As was previously mentioned the main problem in the series connection of IGBTs is the sharing of the voltage across the IGBTs. The snubber operates during switch-off. The snubber limits the voltage across individual IGBTs not to exceed the device's voltage limit. The snubbers were designed to start operating at 800 V per IGBT. Figure 4-9 shows the voltage waveforms at the nodes between the individual IGBTs during the switch-off transient. The voltage does not share equally during switch-off, but the IGBTs to switch off first takes most of the voltage. In Figure 4-9 voltage is mainly taken by IGBT numbers 1, 2, 5, 6 and 7. IGBT numbers 1, 5, 6 and 7 are at their limit according to their respective snubbers and Number 2 takes the remaining voltage. If the voltage were to rise further the other IGBTs would also start taking voltage.

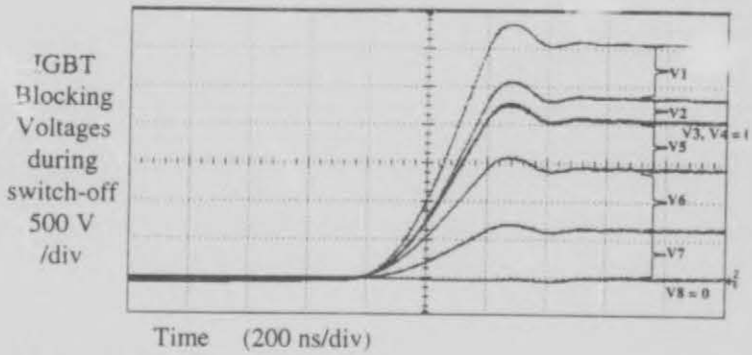


Figure 4-9 : IGBT voltage waveforms during switch-off

After the switching instant the sharing resistors start dividing the voltage between the IGBTs equally. The speed of this action is determined by the RC time constant between the sharing resistors and the snubber capacitors. Figure 4-10 shows how the voltages start unequal after the switch-off transient and end nearly equal at switch-on.

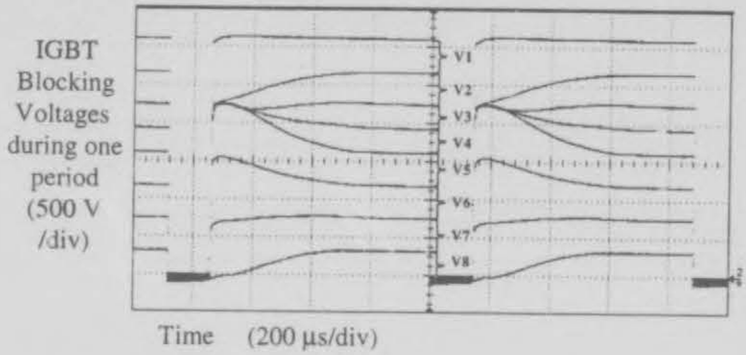


Figure 4-10 : Node voltages in the IGBT string during one period

At the time the IGBTs switch on again, the voltage across them is equal. The switch-on transient is shown in Figure 4-11. The on-state voltage across the string of IGBTs is about 25 V. The IGBTs switch from 3 kV dc to 25 V in about 500 ns. This causes severe electromagnetic interference in all electronic circuitry. It can be noted node voltages 4 and 5 rise before they drop to zero. This is caused by IGBTs 1 to 3 and 6 to 8 switching on before 4 and 5. The voltage becomes lower across the switches that switch first, while those that have not yet switched has to take the voltage. When these also start to conduct the voltage appears across the dumping resistor.

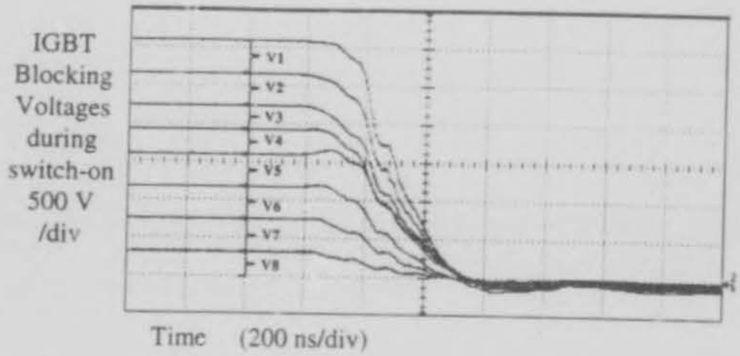


Figure 4-11 : Node voltage waveforms in the IGBT string during switch-on

It has been said that the resistor is very inductive. This can be seen in Figure 4-12 where the current through the inductor is shown with the voltage across the IGBT switch. When the switch is conducting the voltage across it drops to zero but, because of the inductance of the resistor bank, the current does not immediately jump to its final value. The time it takes to reach two thirds of its final value is about 300 µs. With a resistive value of 60 Ω this implies an inductance of about 18 mH. This is considerable when switching at 1 kHz.

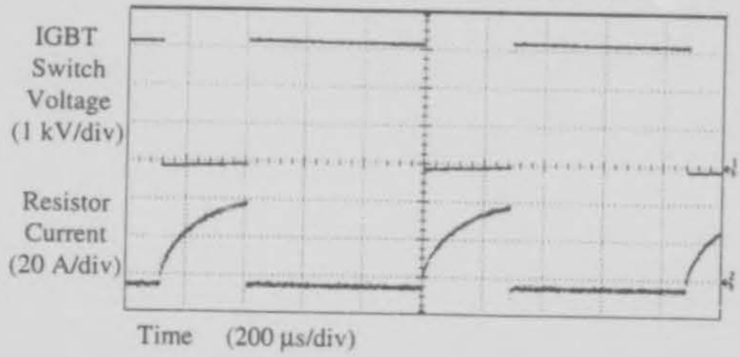


Figure 4-12: Resistor current vs. IGBT switch voltage

The commutation of current from the IGBTs to the freewheel diode can be seen in Figure 4-13. When the IGBTs are conducting the current flows through them and none through the freewheel diodes. When the IGBTs stop conducting the current commutates from the IGBTs to the diodes and then reduces to zero as the stored energy is dissipated in the dump resistor.

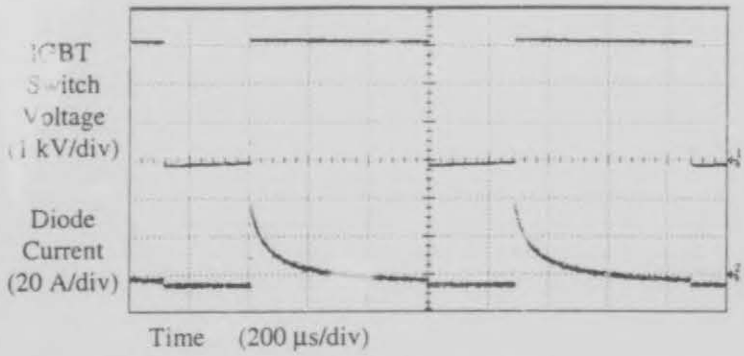


Figure 4-13 : Freewheel diode current vs. IGBT switch voltage

The last consideration in the operation of the solid state dc dump, is the sharing across the freewheel diode during the time that it has to be blocking. If this does not operate correctly the diodes will be destroyed by the over-voltage. The diodes will the open circuit permanently and the freewheel operation will not operate. The IGBTs will then be destroyed by the over-voltage caused by the inductance of the resistor bank. The voltage sharing across the diodes is shown in Figure 4-14. Again the voltage sharing is not perfect but with a diode rating of 1700 V blocking voltage it is sufficient.

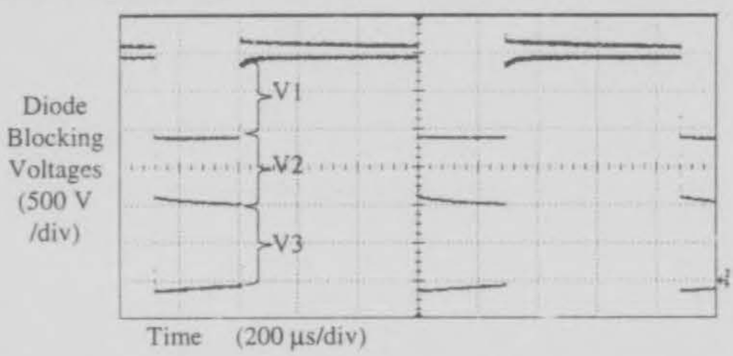


Figure 4-14 : Voltage sharing across the freewheel diodes

4.4.2 PI Control

In the operation of the dc dump it is important to see that the voltage does not rise exorbitantly above the pre-set value of the dc dump. This is the main function that the dc dump has to perform. It is also important that the dump react quickly to system variation to protect the other system components.

In the testing of the PI controller a limit of 3 kV was set on the dc bus value. For safety reasons the rest of the system was not connected in the testing phase of the dc dump. The testing of the dc dump consisted of using the soft-starter to control the dc bus voltage (By varying the firing angle) and monitoring the response of the converter. Because of the physical separation of the driving circuitry and the converter the high voltage and low voltage signals could not be combined onto a single trace. The method of testing the PI controller was to have the dc bus voltage steady at a voltage below the pre-set value of the dc dump. A step change in reference for the soft-starter is then given to have the bus voltage rise above the specified limit.

Figure 4-15 shows the voltages inside the analogue PI controller. The reference voltage to the soft-starter is first 8 V which corresponds to 1.5 kV on the dc bus. The reference is then dropped to 5.7 V which , without the working of the dc dump, would charge the dc bus to 3.8 kV. At the point in time marked X the voltage rises through the 3 kV pre-set limit of the dc dump. The pwm duty cycle is changed to limit the dc bus voltage. The dc supply has inductance in the path and therefore does not rise with a step response according to the step reference given to the soft-starter. It is also non-linear and does not give the classical second order response. Yet the working of the controller can be seen to be effective.

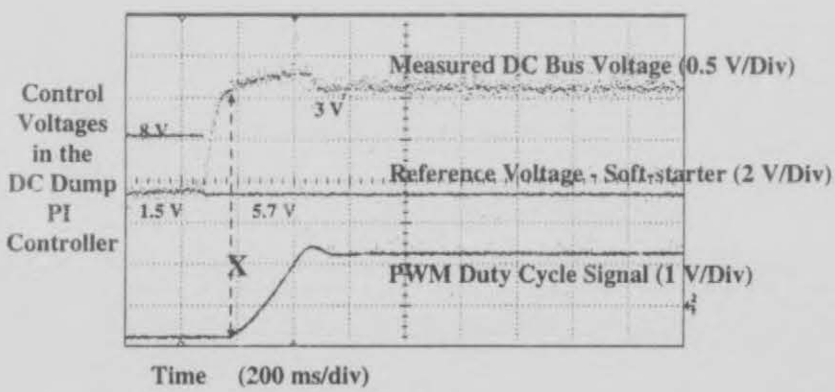


Figure 4-15 : Reaction of the dc dump controller under a simulated fault condition

The most important measurement is not that of the controller, but that of the actual dc bus. It is here where the voltage needs to be limited to prevent over-voltage damage. A high voltage probe was connected to the dc bus and the measured voltage is shown in Figure 4-16. It can be seen that the voltage rises 200 V above the limit before it is brought back to 3 kV. This represents a 6.7 % overshoot. The time it takes to limit the voltage is about 300 ms.

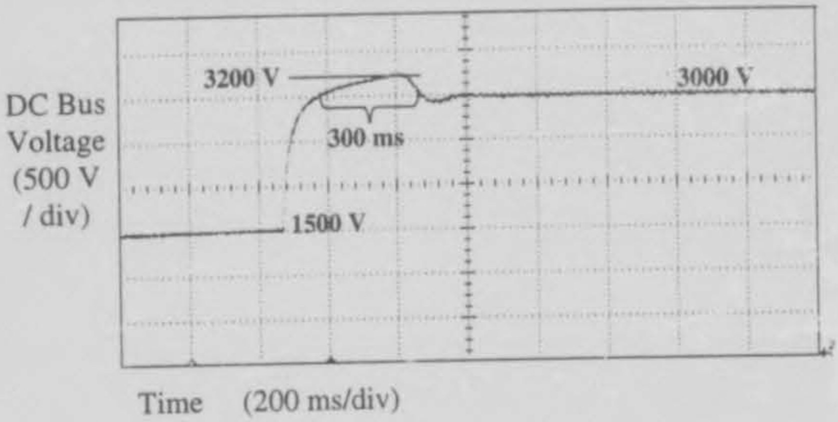


Figure 4-16 : Actual dc bus voltage during over-voltage limiting

When the voltage drops back to 3 kV it shoots over slightly below 3 kV before settling at 3 kV. The controller could be made to operate faster, but the oscillations would increase until the system becomes unstable.

4.5 Conclusions

In the development of the solid state dc dump switch certain facts were clearly shown.

- In the series connection of the IGBT devices voltage sharing was the main problem. Using advanced over-voltage snubbers, this problem was effectively addressed. The switch has only been implemented for 50 A. Using larger IGBTs and snubber circuits this could be expanded to switch up to 1000 A which would account for an entire substation's regenerative braking
- The devices have been designed for the absolute worst case scenario. If the parameters of the line and resistor banks are measured the devices could be redesigned considerably smaller according to the actual system parameters.

- The measurement of the dc bus voltage is currently being done by a resistive divider and fed directly into the controlling electronics. This is very effective and gives a good signal, but it has two distinct disadvantages. The one is that in the case of isolation failure the measurement is electrically connected to the controller. This has the danger that the controller could be damaged or destroyed. The other disadvantage of the electrical connection is that the ground of the power and the controlling circuitry has to be the same, which normally is not the case. An isolation interface would have to be developed for the voltage measurement.
- The implementation of PI Control in the operation of the dc dump has proven to be very effective. The voltage is clamped to only a 6 % overshoot within 300 ms of over-voltage. The response is not a second order graph because of the non-linear nature of the operation of the soft-starter.

Chapter 5

Development of the Fuzzy Controller

5. Development of the Fuzzy Controller

5.1 Introduction

Fuzzy control has been used extensively in industry for a variety of processes [M10]. These include controlling robotic motion [A22] to the stabilising of a power system [B15]. To the authors knowledge it has been used once in the South African traction system [A23]. The principle of intelligent system protection is to monitor the system condition and to have the system respond in such a way as to protect itself. The way this fuzzy controller does this is by evaluating all the system parameters, like dc bus voltages and the current drawn from the ac grid, and controlling the system inputs to keep these parameters within safe limits. The steps of fuzzy control is shown in Figure 5-1.

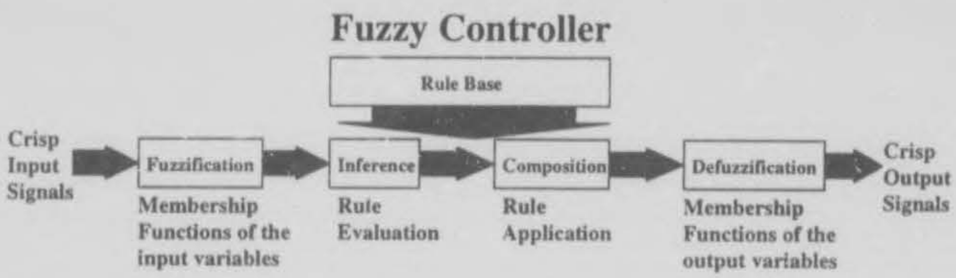


Figure 5-1 : Procedure inside a fuzzy controller

The concept of fuzzy control is to evaluate inputs according to certain criteria. The input then loses its crisp value to a set of functions that describe that input. None of the functions have the absolute description of the variable, but each weighs the variable against that certain criterion. More will be said about that later. The rule base control the way the controller react to its inputs and control the outputs. The output of the fuzzy controller also needs to be a crisp value. The fuzzy values inside the controller need to be translated to crisp values that can be reacted to by the rest of the system.

The operation of the fuzzy controller will first be demonstrated in its operation of the dc dump. The fuzzy controller will then be expanded to the operation of the whole prototype system.

5.2 Fuzzification

The step of fuzzification is the process where crisply defined input values are translated to a set of values that characterise that input. These values are determined by the rule base that says that for a certain characteristic of the input, the output would have another characteristic. For this reason the fuzzification step and the rule base are usually developed together. The dc dump is a voltage controlled device so the voltage needs to be characterised linguistically for the fuzzy controller.

The main dc bus is expected to vary to some degree about a working voltage of about 3.3 kV. This is because of the impedance of the transformer and the regenerative action of the dc motor drive. Two error conditions can occur. The voltage can be pulled too low by excessive powering or the voltage can rise too high due to excessive regeneration. A range of variation is allowed of about 200 V. Figure 5-2 shows the functions describing the main dc bus voltage.

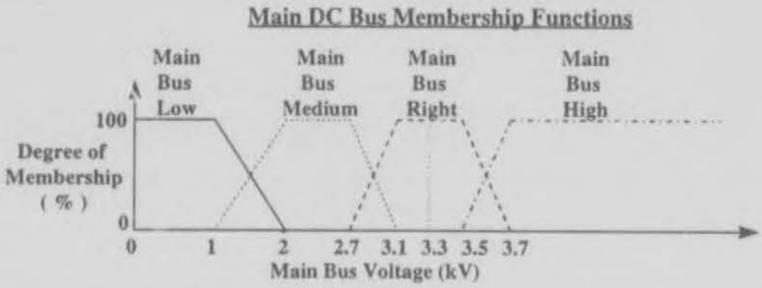


Figure 5-2 : Membership functions for the main dc bus voltage

It can be seen that, if the voltage drops lower then 3.1 kV, the controller starts to see it as an error and starts compensation. When the voltage reaches 2.7 kV the controller sees the error as critical and does maximum compensation for whatever could be pulling the dc bus down. Similarly, if the dc bus rises above 3.5 kV, the controller sees a small error condition. It will then start compensating using the dc dump. If the bus keeps on rising the error becomes more critical until at 3.7 kV the error is at maximum. After the highest level is breached, back-up measures will have to be taken.

The lower part of the voltage range is divided up into two sections. This is to distinguish between start-up where the voltage is very low, and the condition where the voltage is pulled low. If the voltage is pulled too low and starts to enter the lowest part of the range, then the controller can know that the compensation of the fuzzy stage is not working sufficiently and the system should shut itself down.

5.3 Rule Base

The rule base is the part of the controller where the intelligence of an experienced operator is implemented in the computer. The rule base could be described as the "intelligence" behind the fuzzy controller. The rule base is a collection of rules that determine the controller's response to the weighted inputs to the fuzzy controller. In this specific system there is three outputs that the fuzzy controller can influence to protect the system : the duty cycle of the dc dump, the speed and the torque on the rotor. In the operation of the dc dump only the duty cycle is of importance. The dc dump is only controlled by the dc bus voltage so the rule base for its operation is very simple :

- Rule 1 : IF[1] the **Main Bus**[2] is **HIGH**[3] THEN
  Duty Cycle is **HIGH**[4]
- Rule 2 : IF **Main Bus** is **NOT HIGH** THEN...}
  Duty Cycle is **LOW**

These rules linguistically describe the operation of the dc dump. In the fuzzification stage the Main Bus has been linguistically defined using certain membership functions. The linguistic variables **HIGH** and **LOW** need to be defined as membership functions as well. This is necessary to convert the output (The duty cycle) back to a crisp value that the converter can operate on.

5.4 Inference

The inference stage is a process where the controller takes the weighted inputs supplied by the fuzzification stage and uses them to determine the "truth" of each rule. Each rules is assigned

---

[1] Operators are written in Normal CAPITAL LETTERS

[2] Input and Output parameters are written in **Normal Bold Letters**

[3] Membership functions of inputs and outputs are written in **BOLD CAPITAL LETTERS**

[4] The Delphi code for implementing these rules is shown in Appendix D.

a percentage of truth according to the truth of its argument. The standard definitions for the AND, OR and NOT operators in fuzzy logic are:

truth (not x) = 1.0 - truth (x)

truth (x and y) = minimum (truth(x), truth(y))

truth (x or y) = maximum (truth(x), truth(y))

Using this methodology the truth of the each rule can be determined as a percentage. When the truth of the specific rule is determined it is applied to the output membership function of the rule using one of two methods. The one method in called the minimum method where the output membership function is clipped to the value of the truth of the rule. This is called MIN inferencing. The other method is called the PRODUCT method where the output membership function is scaled by the value of the truth of the rule. In this controller the product inferencing method was used.

The controlling output of the fuzzy protection controller is duty cycle of the solid state dc dump. The duty cycle is characterised by a two membership functions being High and Low. These membership functions are shown in Figure 5-3. This linear function is the simplest form of membership function and more complicated methods exist. For this first prototype these were found to be sufficient.

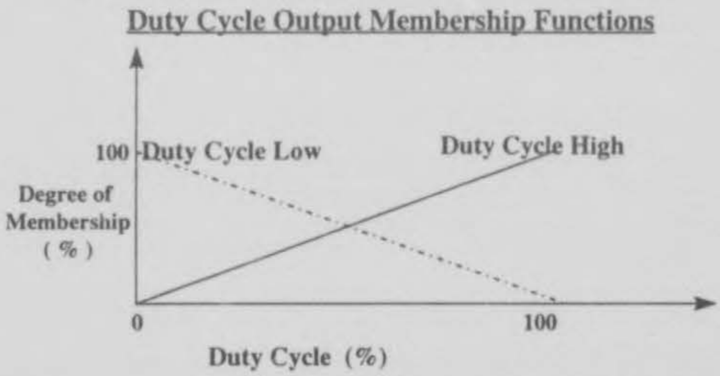
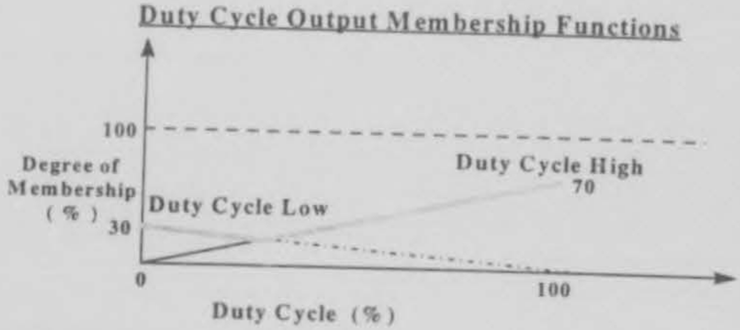


Figure 5-3 : Membership functions describing the duty cycle output

5.5 Composition

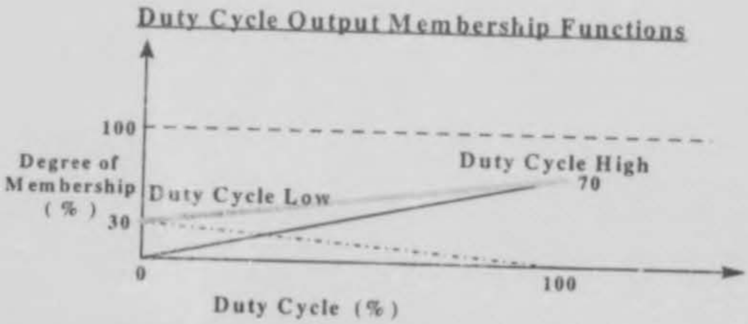
The composition stage is where the output membership functions of all the rules that are applicable to a certain output are combined to form a single output membership function for that output. There are also two main methods of combining the individual membership functions to form a single membership function. These are the MAX and SUM methods. The MAX method takes the pointwise maximum over all the separate membership functions to construct the final membership function. Figure 5-4 shows the deriving of the final membership function using the truth values shown in the box and the MAX composition method.



Duty Cycle Low	- 30 %
Duty Cycle High	- 70 %

Figure 5-4 : Final membership function for duty cycle using MAX composition

The SUM method summates all the individual membership functions and then is normalised to 100 percent again. This is demonstrated in Figure 5-5 using the same values as was used for MAX composition. The SUM composition method can give truth values greater than 100 % which has to be normalised to give a maximum of 100 % again.



Duty Cycle Low	- 30 %
Duty Cycle High	- 70 %

Figure 5-5: Final membership function for duty cycle using SUM composition

In this application is was decided to use MAX composition for the final membership function describing the duty cycle. The membership function is still a fuzzy linguistic variable and needs to be converted to a crisp value (Voltage) that can drive the converter.

5.6 Defuzzification

The final step in the fuzzy controller is converting the output membership function, which is a fuzzy set, to a crisp value that is useful to the rest of the system. For the dc dump the control signals that need to be defuzzified is the duty cycle. There are several methods on defuzzifying fuzzy sets. Two of these methods are Centre Of Maxima (COM) defuzzification and Centroid method defuzzification.

5.6.1 Centroid Defuzzification

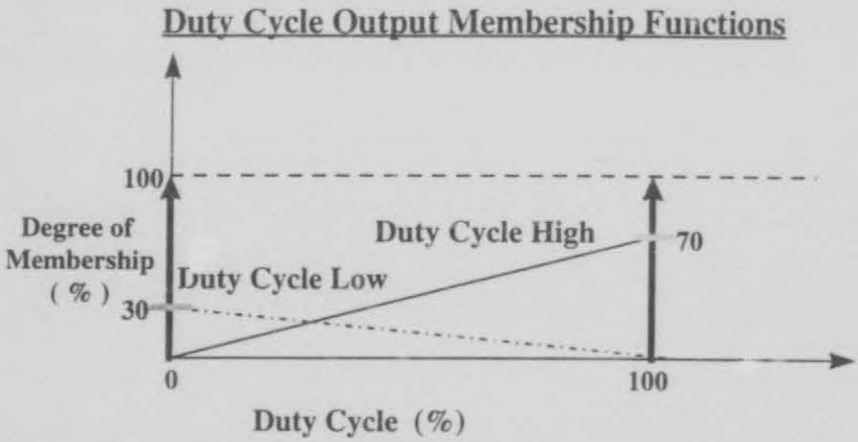
In the Centroid method, the crisp value of the output variable is computed by finding the variable value of the centre of gravity of the membership function for the fuzzy value ([T10], [M11]). In this application MAX composition was used so in the defuzzification stage there will be referred to Figure 5-4 as the final membership function for the duty cycle. The method to determine the centre of gravity is simply by calculating the moment of the function divided by the area of the function.

In Figure 5-5 it can be seen that the variation in the values of the duty cycle is limited to a certain range (0 to 100 %). If this is taken as the universe of discourse [M11], the area can be calculated by integrating f(x) dx, where f(x) is the membership function and x is the output variable. The moment of the function is calculated by integrating over the same range x * f(x) dx. The result will be a value inside the range defined for the variable. This crisp value (Between 0 and 100) will be used as an output to the rest of the system.

5.6.2 Centre of Maxima Defuzzification

The Centre Of Maxima method computes a crisp output as a weighted mean of the term membership maxima, weighted by the inference results. The locations of the individual term membership maxima are indicated by the arrows in Figure 5-6 and the inference results are shown by the crossbars on each arrow. Let V be the linguistic variable (Duty Cycle) to be defuzzified, let μ_{V_i} be all the membership functions of all linguistic terms i defined for the variable to be defuzzified, and let μ_{Li} be the inference result for every linguistic term i. The crisp output value v for the fuzzy variable V is defined by the following equation:

$$v = \frac{\sum\left[\mu_{Li}.\max(\mu_{Vi}).\arg\left(\max(\mu_{Vi})\right)\right]}{\sum \mu_{Li}} \quad \text{.........(1)}$$



Duty Cycle Low	- 30 %
Duty Cycle High	- 70 %

Figure 5-6 : COM defuzzification of the torque membership function

In the dc dump controller the COM method of defuzzification was used to calculate the duty cycle of the 1 kHz pwm signal driving the converter.

5.7 Practical Results of the dc dump using Fuzzy Logic Control

The criteria for the operation of the dc dump has been discussed in the previous chapter. The same measurement system and pwm generator has been used for both the controllers. Only the control block of the analogue PI controller has been replaced by the PC employing fuzzy control.

The same method of creating a step in the dc bus voltage is used to force the voltage to go over the pre-set limit of 3 kV. The response of the control signal is shown in Figure 5-7.

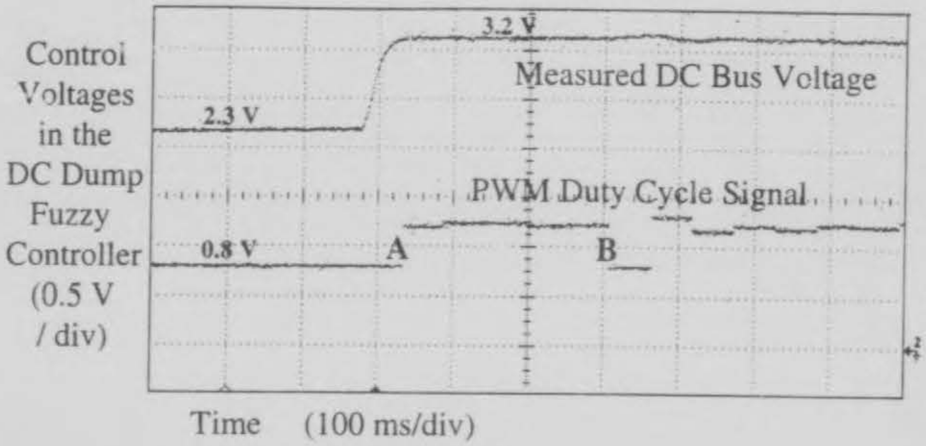


Figure 5-7 : Reaction of the fuzzy logic dc dump controller under a simulated fault condition

The duty cycle of the converter is generated using a UC3524 chip. The duty cycle does not start increasing immediately as the input voltage rises above 0 V. The output of the fuzzy controller is set to have a minimum value of 0.8 V which is the maximum voltage which still enforces a zero duty cycle. In the voltage waveform of the duty cycle reference signal the cycle time of the fuzzy controller can be seen. It takes about 50 ms for the cycle of measurement, calculation and output to be completed. In Figure 5-7 the first calculation of duty cycle after the error (Point A) is very close to the final value for compensation and the voltage stabilises very quickly. Due to an error in measurement (Due to noise - Point B) the controller writes out a wrong output for one cycle. The voltage then rises a bit and the controller has to compensate. It can be seen that the system takes about 100 ms to settle again. In Figure 5-8 the actual dc bus voltage was measured again under fuzzy logic control. The fast response of the controller can be seen in the small over-shoot in the voltage. Again an error in measurement caused an incorrect control signal to be written out. This time the voltage was pulled exorbitantly low, but once again it took about 100 ms to return to its stable value.

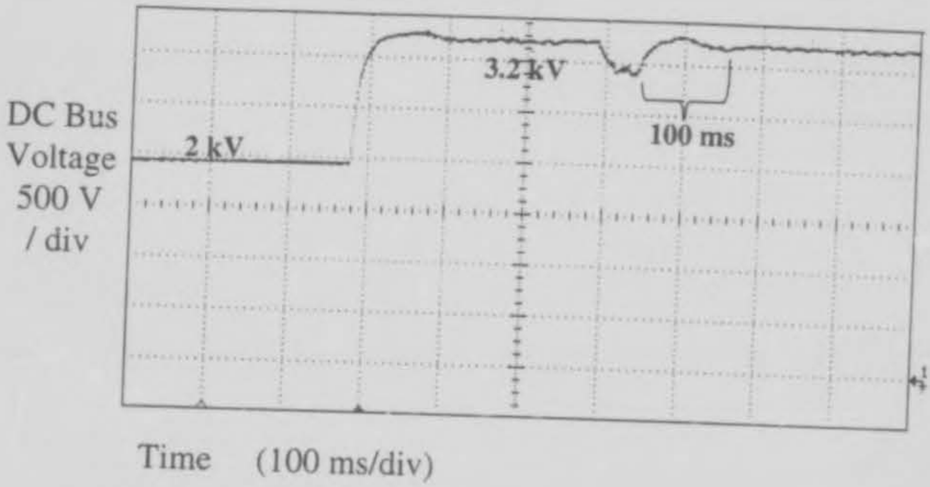


Figure 5-8 : Actual dc bus voltage during over-voltage limiting

The reaction time of the fuzzy controller is about 3 times faster than the PI controller of the previous chapter. The fuzzy controller that has been employed does not have an integrating function. This means that it will have a constant following error contrary to the PI controller. A digital integrator can be implemented to address this problem.

The concept of fuzzy control could easily be expanded to the control of the rest of the system.

5.8 Conclusions

A controller has been implemented in a prototype system representing a practically large system. The role of this controller is firstly system protection and secondly interfacing with the system operator to facilitate the system operation. The fuzzy controller monitors all the system parameters and implements the intelligence of an experienced system operator in software format. The system evaluates the inputs and makes intelligent control decisions according to the rules set by the operator. The controller then interprets the results of the applied rules to give outputs that are of meaning to the rest of the system.

Chapter 6

Intelligent System Protection

6. Intelligent System Protection

6.1 Introduction

All systems used in industry applications have an area of safe operation. This might vary in complexity from a simple motor drive that is limited to a certain maximum speed, to complex systems that have numerous currents, voltages, speeds, etc. that cannot exceed specified limits for each parameter. In a typical system the protection scheme will prevent the system from exceeding its specified limits. An error would cause a shutdown of the appropriate subsystem or the total system.

In chapter 3 the protection in a typical Spoornet substation is discussed. The basic principle is that a device monitors a parameter and shuts down if there is an error. The principle of intelligent system protection is that parameters from all over the system is monitored by a simple controller. The controller will then make appropriate decisions to handle the error by controlling the system as a whole instead of just localised protection.

In this chapter two methods of intelligent control are discussed and certain conclusions made. The one method is a shutdown controller and the other a fuzzy controller that influences the system inputs. The one could be operated without the other, but for maximum safety it was decided not to operate the fuzzy controller without the shutdown controller as a backup. Figure 6-1 shows the relation of the protection systems to the area of normal operation. The shutdown level of error can only be zero or one. This implies that the response to any fault condition will be totally binary.

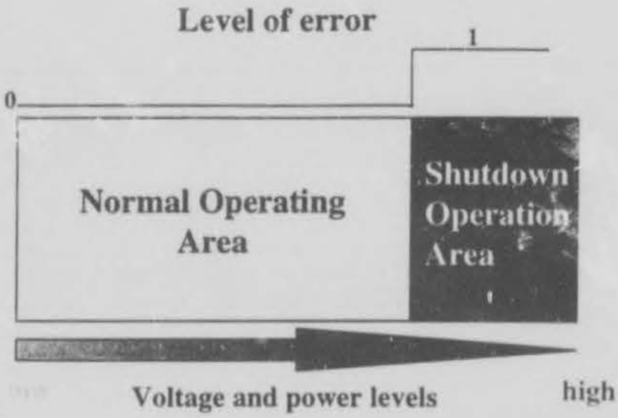


Figure 6-1 : Shutdown Control Layers of Protection

A fuzzy controller is proposed that acts as a buffer between the normal operating area where there is no errors and the shutdown area which can be seen as a critical error area (Figure 6-2). The purpose of the fuzzy controller is to evaluate the weight of the error and to scale the response accordingly.

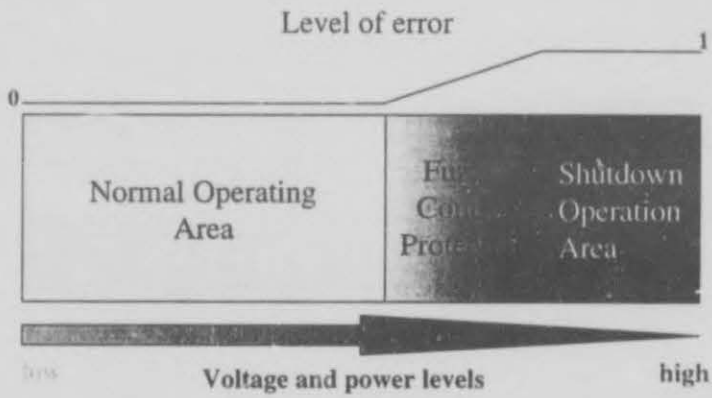


Figure 6-2 : Fuzzy Control Layers of protection

6.2 Shutdown System Protection

The conventional protection scheme monitors the parameters of the system and responds by shutting down either a part of the system or the whole system. This is shown as two layers of operation in Figure 6-1. In the following section a protection controller employing fuzzy control is discussed. The inputs to the fuzzy system controller and the shutdown controller are identical and are summarised in Table 6-1.

Table 5-1 : Protection input parameters [T11]

System Parameter	Operating Area	Highest Limit
Phase Input Currents	< 150 A	200 A
Main dc Bus Voltage	3.3 kV	4 kV
AC-AC Converter DC Bus Voltage	800 V	850V
Anti-Parallel DC Bus Voltage	800 V	850 V
Anti-Parallel Current	50 A	70 A
Speed	< 1440 rpm	1500 rpm

The strategy of error handling employed by the shutdown layer is to isolate the error from the rest of the system by switching off the surrounding subsystems. The different subsystems are shown in Figure 6-3 with each block addressed as a separate unit.

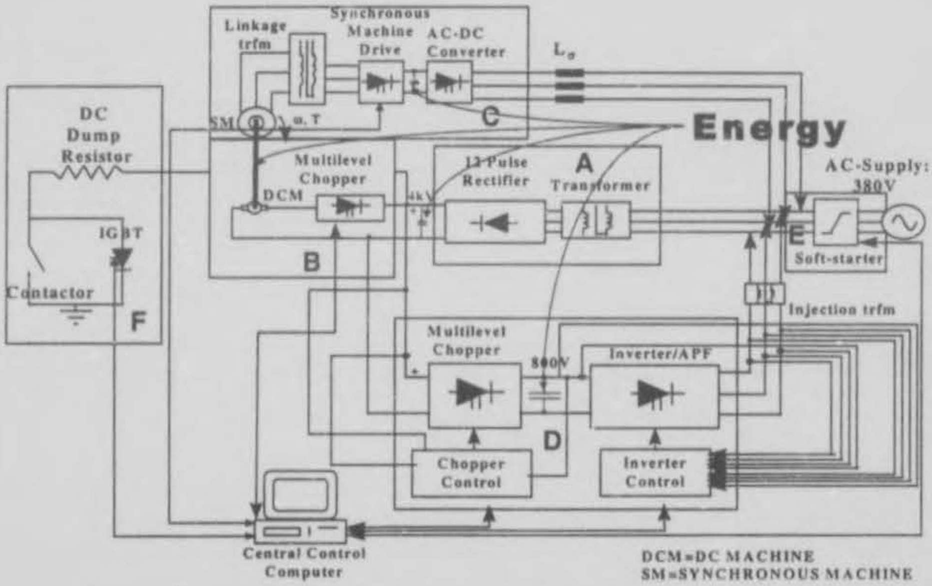


Figure 6-3 : Traction substation prototype system

Energy is stored throughout the system in capacitive, inductive and kinetic form. The shutdown protection strategy will concentrate on the places where large amounts of energy are stored. These are the charges stored in the three dc buses and the kinetic energy stored in the rotor of the motor.

6.2.1 Energy Stored in DC Buses

The energy stored in the dc buses can easily be calculated if the capacitance value and the voltage is known.

$E_{stored} = 0.5 \times C \times V^2$ .... (1)

The three buses are the main 3.3 kV bus, the anti-parallel path dc bus and the ac to ac converter dc bus. The energy stored in the respective buses are as follows.

Main bus :

Voltage : 3.3 kV

Capacitance : 500uF + 400uF + 2 640uF = 3 540uF

Energy : $0.5 \times 3\,540e\text{-}6 \times 3\,300^2 = 19.275$ kJ

Anti-parallel Bus:

Voltage : 800 V

Capacitance : 26.4 mF

Energy : $0.5 \times 26.4e\text{-}3 \times 800^2 = 8.448$ kJ

AC to ac converter Bus:

Voltage : 900 V

Capacitance : 13 mF

Energy : $0.5 \times 13\ e\text{-}3 \times 900^2 = 5.265$ kJ

Total energy stored in dc buses = 33 kJ

6.2.2 Energy Stored in the Motor

The other main storage element in the system is the rotor of the two motors. The energy stored in the rotor has to be dissipated for the motor to stop and the system to be safe. The inertia of the rotor can be calculated using the rundown test [T12], Appendix E on the rotor. The inertia of the rotor was measured to be 23 kg.m^2. The energy stored in the rotor during maximum speed is then

$E_{stored} = 0.5 \times J \times \omega^2$ ....(2)

with ω the radial speed of the rotor. The maximum speed of the motor is 1500 rpm which translates to 157 rad/s. The energy stored in the rotor is then 284 kJ.

The energy stored in the rotor is many orders of magnitude greater than that stored in the dc buses. This does not make the other places of energy storage insignificant. The dc buses are connected to IGBT switch-gear that are very sensitive to over-voltage [T13]. The distribution of stored energy will determine the strategy in the shutdown procedure of the system controller. The motor has to be stopped before the dc buses can be discharged and all the subsystems be switched off.

6.2.3 Shutdown Protection Strategy

As mentioned earlier the basic strategy employed is shutting down only the necessary subsystems to isolate any fault that has occurred. A communication protocol is established between the Central Control Computer (CCC) and the subsystems to give the CCC control over the operation of the system. The communication with the subsystems are summarised in Table 6-2. In such a noise intensive area as a substation a controller malfunction has to be kept in mind. In the event of a controller malfunction the conventional protection would still be there to protect the system hardware, but the advantages of the system controller would clearly be lost.

Table 6-2 : Communication with the individual subsystems

Subsystem	Subsystem to CCC	CCC to Subsystem
DC motor drive	DC motor drive error signal	DC motor drive status
		Speed
Synchronous motor drive	Synchronous motor drive error signal	Synchronous motor drive status
		Torque
Anti-parallel chopper	Anti-parallel chopper error signal	Anti-parallel chopper status
Anti-parallel inverter	Anti-parallel inverter error signal	Anti-parallel inverter status
Soft-starter	Soft-starter error signal	Soft-starter status

An Altera™ EPLD was used to monitor the error signals from all the subsystem and to write the status signals to each subsystem (Appendix A8, A9). It also functions to control the speed and torque signals to the two motor drives during fault conditions. The control of the reference signals is to ensure controlled operation even during switch-off of the system.

Two classes of errors were identified. The one class is subsystem errors where the error signal of a subsystem would indicate that something inside that subsystem is faulty. The other class is the parametric errors such as over-voltage on one of the dc buses. These parameters are listed in Table 6-1.

6.2.4 Shutdown Rule Base

For the application of the above strategy a rule base had to be constructed. This would be applied by the EPLD to control the protection of the entire system. The first rule base is that necessary for parametric errors. If a parameter goes out of bounds all systems that will be affected by that error, or could have caused the error, are switched off. The rules for the parametric errors are listed in Table 6-3. The code for this implementation is shown in Appendix F.

Table 6-3 : Parametric errors and system response

Parametric error	System response
System over-current	Parallel inverter off {No current drawn through anti-parallel path} Parallel chopper off {No current drawn through anti-parallel path } Soft-starter off {System isolated from grid} Dc dump on {Sink stored energy of buses and rotor} Speed signal zero {Use dc drive to brake the rotor} Torque signal maximum negative {Use Synchronous drive to brake the rotor}
Synchronous motor drive dc bus over-voltage	Synchronous motor drive off {dc bus isolated}
Anti-parallel dc bus over-voltage	Parallel inverter off {dc bus isolated} Parallel chopper off { dc bus isolated }

Main dc bus over-voltage	Dc dump on {Sink energy lifting the dc bus}
Anti-parallel over-current	Parallel inverter off {No more current in the path } Parallel chopper off {No more current in the path } Synchronous motor drive off {No current injected in grid point} Soft-starter off {No current injected in grid point} Dc dump on {Sink energy stored in system} Speed signal zero {Use dc drive to brake the rotor}

It can be seen that for some errors only one sub-system is switched off while for others the entire system needs to be switched off. Even if the whole system needs to be switched off, the switch-off process can be controlled and the critical places of energy storage can be discharged before the system switches off completely. This is an example of intelligent control. The intelligence of an expert system operator is programmed into the system controller that does the protection control automatically thereafter.

A similar strategy is followed in designing the system responses to subsystem errors. If a subsystem fails, the subsystem needs to be isolated from potentially harmful sources of energy.

Table 6-4 : Rule base for system errors

Error condition	Rule for the response
DC drive error	Parallel inverter off {Regeneration path closed} Parallel chopper off {Regeneration path closed} DC Drive off {Correlate controller with system} Soft-starter off {No extra energy into the system} Dc dump on {Dissipate energy stored in the system} Torque signal maximum negative {Brake motor with the synchronous motor drive}
Parallel Chopper or Parallel Inverter or	Parallel inverter off {Correlate controller with system}

Synchronous motor drive error	Parallel chopper off {Correlate controller with system} Synchronous motor drive off {Correlate controller with system} Soft-starter off {No extra energy into the system} Dc dump on {Dissipate energy stored in the system} Speed signal zero {Brake motor with the dc motor drive}

In this system all the subsystems are inter-linked in some or other way. This means that they can not be totally isolated from each other by switching off the appropriate subsystems. This then necessitates the switch-off of the whole system. In a larger system, where it is possible to isolate the erroneous subsystem, only that section of the system needs to be switched off.

It can be seen that the perspective of the intelligent system protection has moved from local protection to global system monitoring and control. The one weak point in the strategy is that all error conditions is either right or wrong. This implies a large safety margin to ensure against unnecessary system downtime and over-sensitivity for errors.

6.3 Fuzzy System Protection

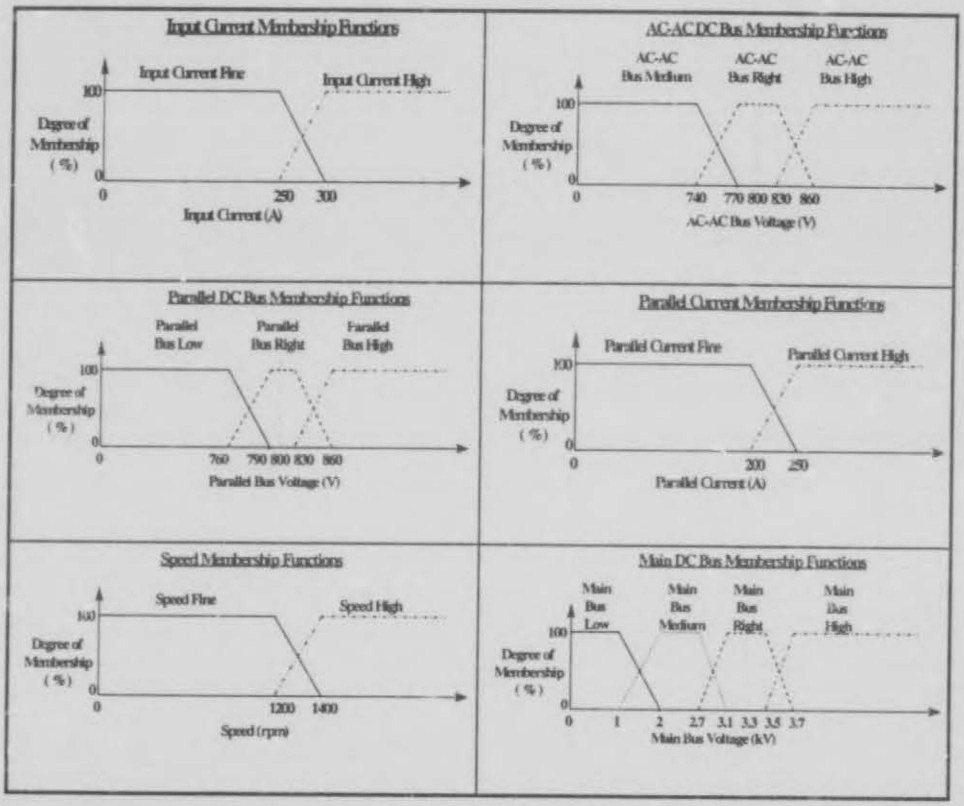
The purpose of fuzzy logic is to model and control systems not according to the on / off logic of the binary system, but to attempt to copy the operation of the human mind. The human mind takes all available criteria and weighs them according to perceived importance. All decisions are then made according to the weight of each input and the effect it has on the current output. The fuzzy control system follows a similar train of thought. Dr. Lotfi Zadeh introduced the mathematics of fuzzy logic in the early sixties [A24].

In the same way as fuzzy logic was applied to the protection of the dc bus voltage, it can also be applied to protect the whole prototype system. The inputs to the fuzzy controller would have to be the same as the shutdown controller, but the control outputs will be the speed reference to the dc drive and the torque reference to the ac drive. This would mean the controller can control the power flow through the system to prevent a serious error condition from occurring.

6.3.1 Fuzzification

The stages inside the fuzzy controller is the same as that of chapter 4. First the inputs need to be fuzzified. The fuzzification of the input variables is summarised in Table 6-5.

Table 6-5 : Fuzzification of system parameters



6.3.2 Fuzzy Calculations

After the inputs have been fuzzified, they need to be applied to a set of rules that govern the operation of the protection controller. These rules need to oversee the protection aspects during start-up as well as during normal operation. The system rule base is summarised in Table 6-6.

Table 6-6 : Rule base for fuzzy system controller

Argument	Statements	Cause
IF[1] **Synchronous Drive Bus**[2] is **LOW**[3] OR **Main Bus** is **HIGH** OR **Parallel Bus** is **HIGH** OR **Parallel Current** is **HIGH** THEN	**Change in Speed** is **LOW** **Torque** applied is **LOW**[1]	Excessive Regeneration
IF **Main Bus** is **MEDIUM** OR **Synchronous Drive Bus** is **HIGH** OR **Input Current** is **HIGH** THEN	**Change in Speed** is **LOW** **Torque Reference** is **LOW**	Excessive Powering
IF **Speed** is **HIGH** THEN	**Speed Reference** is **LOW**	Speed Reference
IF **Input Current** is NOT **HIGH** THEN	**Change in Firing Angle** is **MEDIUM POSITIVE** **Angle output** is **ANGLE CALCULATED**	Start-up
IF **Input Current** is **CRITICALLY HIGH** THEN	**Change in Firing Angle** is **MEDIUM NEGATIVE** **Angle output** is **ZERO**	Over-current
IF **Parallel Bus** is **RIGHT** AND **Synchronous Drive Bus** **RIGHT** AND **Main Bus** NOT **LOW** AND **Main Bus** NOT **MEDIUM** AND **Parallel Current** NOT **HIGH** AND **Input Current** NOT **HIGH** THEN	**Torque Reference** is **USER_ TORQUE** **Change in Speed** is **USER_ SPEED**	Everything Right

These rule determine the response of the controller under all the conditions listed in the right hand column of Table 6-6. The outputs of the system need to be assigned membership functions through the inference and composition stages similar to that done in chapter 4. The Max/Product method was once again used for this stage.

---

[1] Operators are written in Normal CAPITAL LETTERS

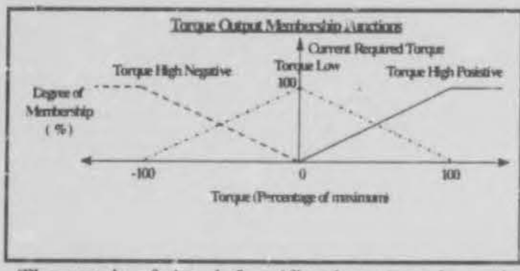
[2] Input and Output parameters are written in **Normal Bold Letters**

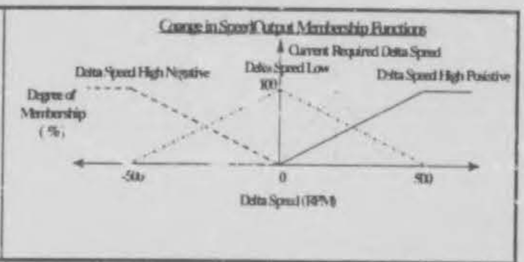
[3] Membership functions of inputs and outputs are written in **BOLD CAPITAL LETTERS**

6.3.3 Defuzzification

The COM method was again employed in the defuzzification of the outputs. The two main controlling outputs are the speed and the torque The membership functions used in the defuzzification stage are summarised in Table 6-7.

Table 6-7 : Output membership functions of the fuzzy system controller





The result of the defuzzification stage is again crisp values that are used as control signal to the rest of the system. These signals prevent excessive amounts of power flowing through the system at any given time and thereby implicitly prevents certain over-voltages and over-currents. The fuzzy controller also controls the start-up procedure of the system.

6.4 Conclusions

The traditional protection scheme uses localised protection devices that each address a specific possible error. This is a very reliable method that has proven its worth over many years. The addition that is proposed is a protection controller that controls the protection from a system point of view. This could enhance the system operation in that it gives the system a method of preventing certain error conditions as well as giving it a measure of ride through capability.

The addition of intelligent protection will protect the system against excessive downtime and/or device over-design. The intelligent control methods are limited in the range of its effective operation. The intelligent protection scheme also employs electronic controllers that operate in a high EMC environment. Due to the relatively low reliability of this application, and its inherent limitations, the intelligent protection controller still needs to be backed up by the conventional protection devices to ensure maximum safety of operation. In this chapter

[1] The Delphi code for implementing these rules is shown in Appendix B.

the shutdown controller was discussed extensively. The next chapter will focus more on the operation of a fuzzy controller that is a further development on the shutdown controller, employing new technology in the protection area.

Chapter 7

Practical Operation of the System Protection Controllers

7. Practical operation of the system protection controllers

7.1 Introduction

In the discussion of the operation of the scale system the focus will not be on the individual operation of all the subsystems as that is not within the scope of this thesis. Rather a systems approach will be followed where the effect of certain error conditions in the system will be studied. The system controller has the dual purpose of system manager and system protector. As manager it has to oversee the start-up procedure of the system. As system protector it has to oversee error handling and shutdown procedures where necessary.

7.2 Start-up Sequence of the Prototype System

The purpose of the start-up procedure is to charge the individual dc buses to their full operational voltages without drawing an in-rush current that damages the system. It will also co-ordinate the switching on of the subsystems until the whole system is operational. A graphic visualisation of this procedure is shown in Figure 7-1.

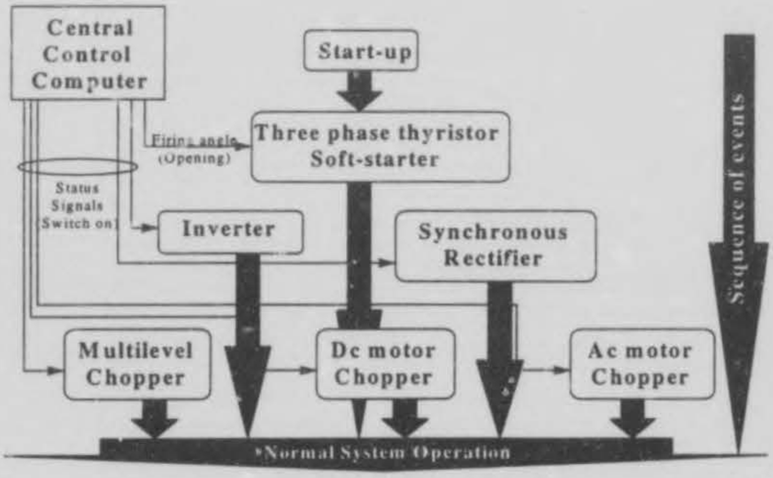


Figure 7-1 : Start up sequence of events for scale system

The start up procedure had to be implemented in both a EPLD and a PC fuzzy controller (The EPLD used an A/D chip and the PC an PC30GA™ A/D card ([T14], [T15], Appendix A10) to interface with the system). The start up sequence will be discussed using the EPLD controller. The first purpose of the controller is to limit the current while charging up the respective buses. The only control that the controller has over the current is through the firing angle of the soft starter. If excessive currents are being drawn the firing angle of the soft-starter is closed. The thyristors will only conduct for a short period of time and the impedance of the magnetic elements (transformers and inductors) will limit the currents drawn during the short conducting time. The firing angle of the soft starter is controlled by a dc voltage varying from 10 V (firing angle completely closed) to 0 V (firing angle completely open - fully on). It can be seen in Figure 7-2 that the soft starter is trying to open , but the currents drawn prevent it from continuing.

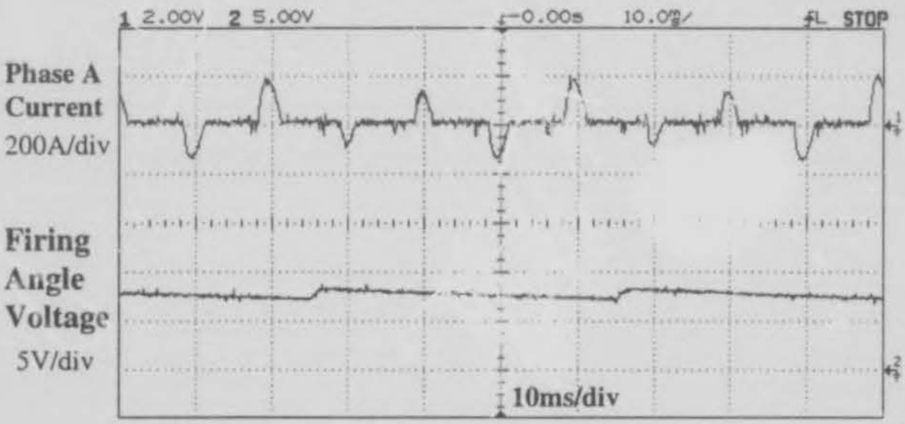


Figure 7-2 : Start up operation of the soft starter

As the capacitors charge up, less current will be drawn and the firing angle will open up till it is completely open. The opening of the firing angle is shown in Figure 7-3 with the corresponding rise in voltage shown in Figure 7-4.

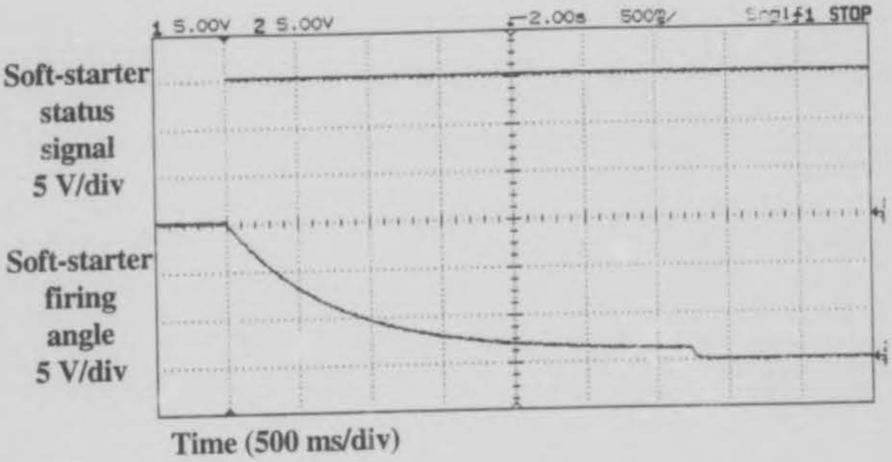


Figure 7-3 : Firing Angle of soft starter during start up

It can be seen that there is not a linear correlation between the firing angle and the voltage, but the correlation is one on one which is enough for protection purposes.

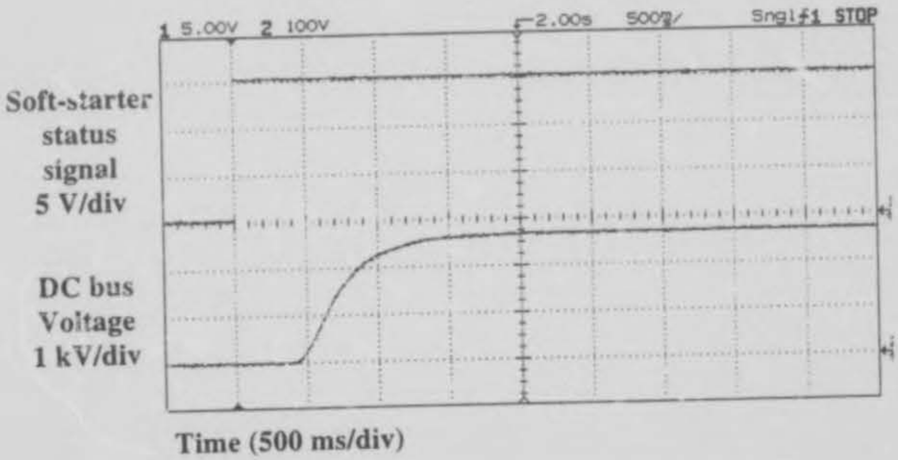


Figure 7-4 : DC Bus voltage during start up

After the angle has completely opened, the status signals of the inverter and synchronous rectifier is set high to charge the remaining dc buses. When these are charged, the two dc to dc converters are switched on and the motor drives' operation enabled.

7.3 Shutdown System Protection

The two methods of protection control is the shutdown system controller and the fuzzy system controller. The shutdown system controller was implemented using both a EPLD and a PC.

The EPLD is an Altera™ device and programmed using AHDL programming language. The code is shown in Appendix F. The shutdown controller has to operate for all possible errors. Using the program simulator the program was tested in an ideal environment. This is shown in Figure 7-5 as a sequence of errors with the system being reset between every error.

- The first error is an 800 V over-voltage error. The parallel inverter and chopper is switched off to isolate the error from the rest of the system. No other part of the system is affected and the rest of the system continues normally.
- The case of over-voltage on the main dc bus is expected (Regeneration) and remedied by switching on the dc dump. All excess regenerated energy will be dissipated in the dump resistors.
- If the dc bus of the ac to ac converter rises unacceptably high the whole synchronous motor drive is switched off and the bus isolated. No load modelling will be possible, but the rest of the system will keep operating.
- In the event of the whole system drawing too much current from the ac grid, the whole system needs to be isolated. This would mean the soft-starter is switched off to prevent any more energy flowing into the system. The parallel path is switched off because it cannot function anymore. The dc dump is switched on to dissipate any energy left stored in the system. The dc motor drive and the synchronous motor drive are both used to brake the motor and send all the energy stored in the rotor to the dc dump.
- If the current in the parallel path rises too high it implies that the dc motor drive regenerates too quickly, the ac drive powers too strongly or there is some error in the parallel path itself that forces the high current. This necessitates an entire system shutdown with only the dc dump dissipating the energy stored in the dc bus. In this case the rotor will run down in its own time according to the friction braking the rotor.

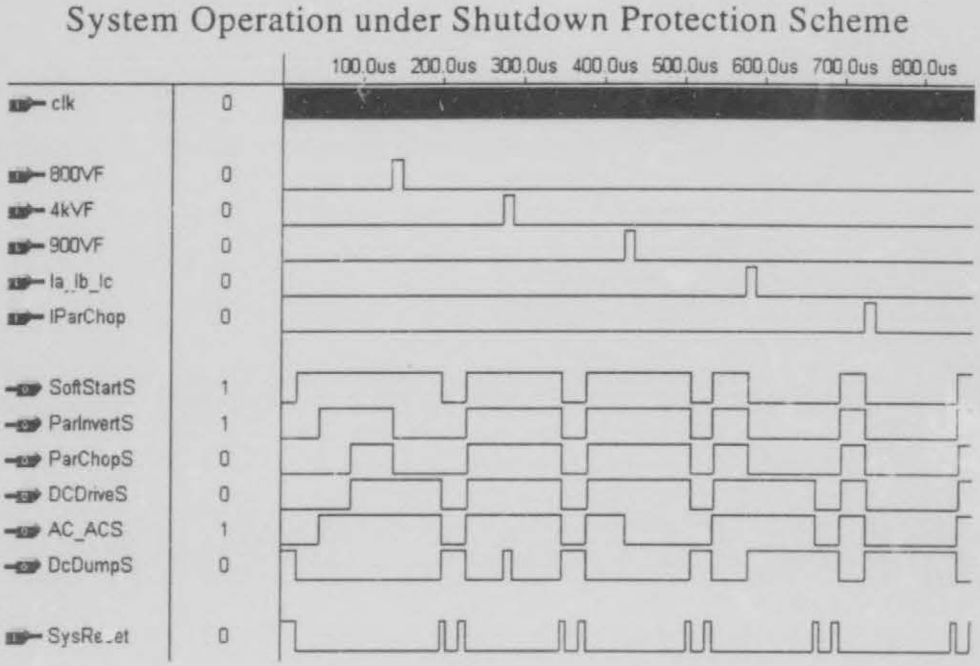


Figure 7-5 : Simulated shutdown controller operation

The reaction of the system controller is determined by that which was programmed into the EPLD. This is simply the experience of a system operator translated to control code. This can easily be modified to refine the operation and to expand the scope (Number of inputs and outputs) of the controller.

The speed of response of the shutdown controller is nearly as fast as its clock speed. A clock speed of 4 MHz was chosen for this controller. An error on the 800 V bus was chosen to demonstrate the response of the shutdown controller. Figure 7-6 shows that the time from the error in the dc bus voltage to the response from the controller is approximately 4 μs.

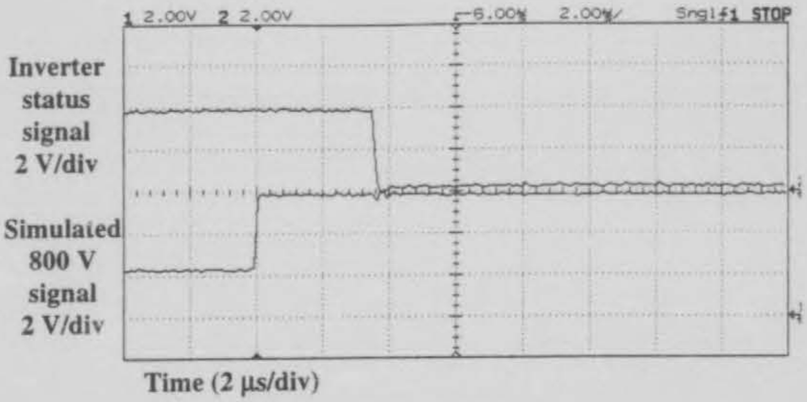


Figure 7-6 : Response time of shutdown controller

The time it would take the system to respond to the control signal from the controller is significantly longer.

7.4 Fuzzy Controlled System Protection

The shutdown controller uses the status signals of the respective subsystems as outputs to protect the system. The output reference signals that the fuzzy controller uses to protect the system are the change in speed signal (accelerate and decelerate) and the torque signal (heaviness of the load). In Figure 7-7 an error condition is simulated with the real controller in operation. The 800 V dc bus voltage is represented by a triangular voltage waveform scaled to a size small enough to be an input to the controller.

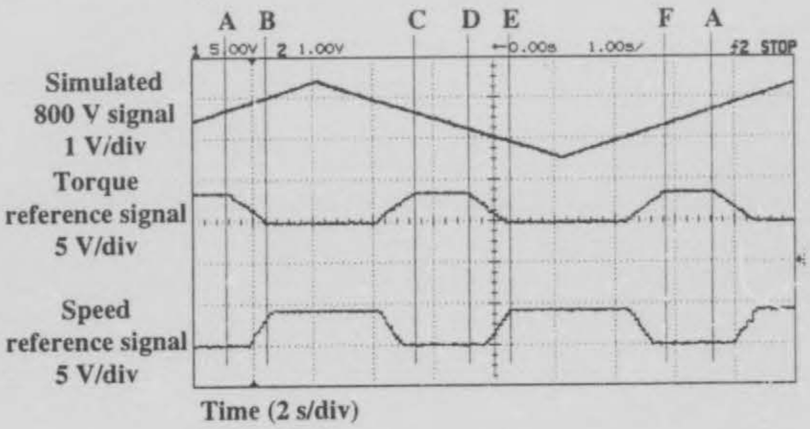


Figure 7-7 : Protective operation of the fuzzy controller

(fo demonstration purposes the digital reference signals were translated to dc voltages. A 10 V signal represents +100 %, a 5 V signal represents 0 % and a 0 V signal represents a -100 % for both the reference signals)

- **A - B** : As the 800 V bus rises through the voltage band (fuzzy band) where the fuzzy controller considers the voltage to be too high (this is determined by the fuzzification process) the controller compensates according to the weight of the error. The deceleration of the rotor and the powering of the synchronous motor drive causes a rise in the parallel dc bus. Limiting the reference signals to the individual subsystems would limit the rise in the bus voltage and the voltage would stabilise at a value inside the fuzzy band. As the voltage rises the compensation increases until the maximum compensation point is reached at **B**. If the voltage rises any further a backup protection scheme has to operate. In this case the shutdown controller was used as the immediate backup to the fuzzy controller with conventional protection devices backing up the intelligent control.
- **B - C** : During this time the voltage returns to its correct operating range.
- **C - D** : This is the range of operation allowed for the parallel dc bus. In this range the reference signals to the system is exactly that required by the user. In this case the torque is 70 % powering and the acceleration -100 % (Maximum regeneration).
- **D - E** : When the bus voltage drops below an acceptable level the controller compensates by dropping the power drawn by the system. This is once again done by decreasing the acceleration and the powering of the respective dc buses. This stops any active power flowing in the system and gives the APF a chance to recharge the dc bus. When the bus is restored to its normal operating voltage the reference signals are restored to their required values.
- **E - F** : The system is restored to normal operation.

The operation of the fuzzy controller is a lot more computationally intensive than the shutdown controller. In this application it was implemented in a PC, but any reasonable micro-controller could be used. The PC uses A-to-D converters to read the analogue input signals, a Pentium 60 motherboard to do the calculations, and D/A converters to write out the reference signals. Figure 7-8 shows the turn around time of the fuzzy controller. From the time the error starts to the output of the protective control signal is about 100 ms.

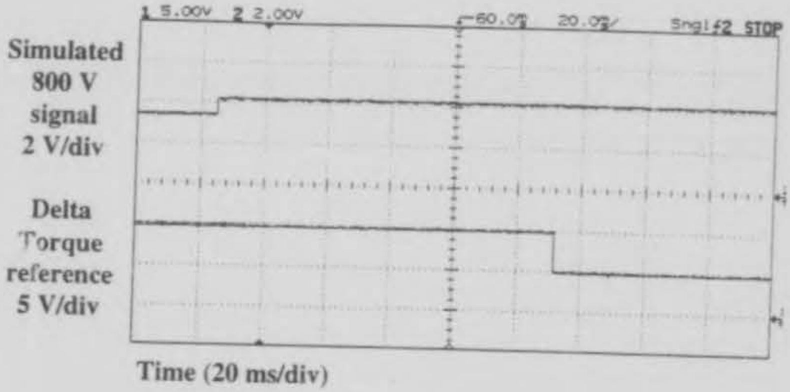


Figure 7-8 : Response time of the fuzzy controller

7.5 Conclusions

In the previous two sections the operation of both protection controllers was viewed using the same error occurring in the system. This is a typical error and representative of all the errors that could occur within the scope of this controller. Because of the similar input to the two controllers the outputs can be compared with regard to certain criteria.

- The first obvious comparison is in the speed of response. Figure 7-6 and Figure 7-8 show the respective response times for the two controllers. The response time of the shutdown controller is about 4 μs and the fuzzy controller about 100 ms. Both are considerably faster than the response of the rest of the system.
- Another comparison can be made in the smoothness of operation. The shutdown controller can only control the system on an on / off basis. If an error occurs the system reacts severely by shutting off the appropriate subsystems. The fuzzy controller scales its response to that of the error. For a small error its compensation is small and vice versa.
- The fuzzy controller has a limited measure of protective capability. The shutdown controller isolates errors from the rest of the system and thereby ensures effective protection. The fuzzy controller needs a backup for severe fault cases.
- The shutdown controller can be implemented in a very low cost EPLD while the fuzzy controller needs the computational capabilities of a small micro-controller.
- The shutdown controller needs extensive programming to facilitate the operation of the system. Any additions to the system is simple in hardware, but difficult in software. The fuzzy controller has a rule base according to which decisions are made. With every

addition of an input, rules governing that input has to be programmed into the controller. The method leans toward modularity and is therefore easy to expand. In both the areas of hardware and software additions to the system are relatively simple.

The above comparisons are shortly summarised in Table 7-1.

Table 7-1 : Shutdown vs. fuzzy protection control

	Fuzzy Control	**Shutdown Control**
Speed	Medium	Fast
Smoothness	Smooth	On / Off
Reliability	Needs Backup	Autonomous
Cost	Medium	Low
Expandability	Relative ease	Easy hardware / Difficult software

Chapter 8

Conclusions

8. Conclusions

This chapter starts with a short overview of the goals that have been accomplished in this thesis. The main conclusions concerning the protection strategies are summarised in section 8.2. The thesis is ended with a look at possibilities of future research to be done as a continuation of this thesis.

8.1 Overview

A 200 kVA prototype system has been developed that represent the dc traction system currently employed by the South African railway company. System protection strategies were studied with both old and new methods described in detail.

- A number of converters have been developed all in the medium power range (Up to 200 kVA). These converters together with some other converters were combined to form a prototype dc traction substation (Figure 8-1). Simple control algorithms have been implemented and tested on these converters in laboratory surroundings. A user / hardware interface has been developed that operate the system while isolated from all high power circuitry.

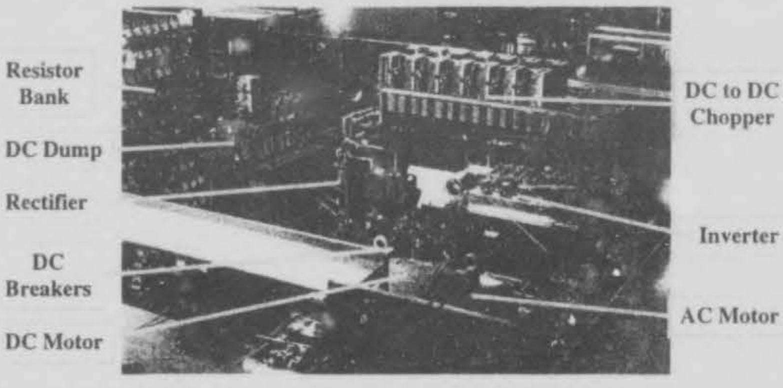


Figure 8-1 : Entire prototype dc traction system

- The protection strategy employed in the current traction system were studied and two new methods of system protection were introduced. The methods were implemented and evaluated in the context of the dc traction prototype system.

Intelligent control was compared to conventional protection strategies with special attention given to ride-through capability, system self protection and cost.

- The one method was implemented in an EPLD with external A/D converters. The EPLD was programmed using AHDL programming language and implemented in controlling the prototype system (Figure 8-2).
- A Delphi application was developed for each of the intelligent protection controllers and implemented using a multi-channel A/D-D/A card (Figure 8-3).

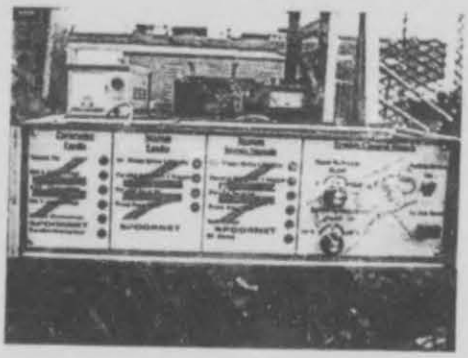


Figure 8-2 : EPLD shutdown controller

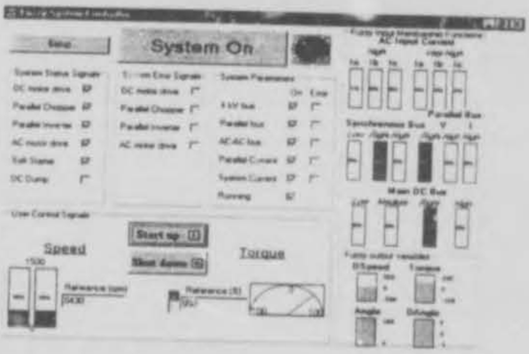


Figure 8-3 : Delphi fuzzy controller screen

- A high voltage dc dump was designed and developed. This switch was used to drive a 100A, 3.3 kV resistor as is used for practical regenerative systems.

8.2 Conclusions

The following are the main facts that have been concluded in this thesis.

1. The main hindrance to the implementation of fast switching silicon technology in the South African traction applications is the high dc blocking voltage that is required. This can be addressed by the series connection of devices (As in the solid-state dc dump), but special care has to be taken to ensure correct voltage division. The implementation of a scaled down system would require a drop in current rather than a drop in voltage to maintain realistic representation.
2. Traditional methods of dc switching using dc contactors have certain drawbacks like high maintenance and lack of smooth operation. This can be addressed using a high voltage IGBT converter employing a pwm switching scheme. PI control has been implemented and proven effective.

3. Conventional protection practices in Spoornet's dc traction substations are effective and relatively fail-safe. Improvements can be made to the protection scheme by integrating intelligent protection functions into the system controller. Protection can be co-ordinated from a system point of view as opposed to a localised protection scheme. Two methods could be used, namely a shutdown system controller or a fuzzy system controller.
4. A shutdown controller is considerably faster than a fuzzy controller, its hardware is easier to implement and costs less. A fuzzy controller gives the system self protecting operation as well as a measure of ride through capability and is easier to expand than a shutdown controller.
5. The parallel path that is proposed has a limited regeneration capability and has to be supported by the conventional resistive brakes. The electrical regeneration system also needs to be backed up by a mechanical braking system.
6. The fuzzy controller is limited in its effectiveness, because it does not have absolute control over the system. In the event of an absolute error like a dead short a controller like the shutdown controller would be necessary that does not try to keep the system operational. It needs to be backed up by another protection system like the shutdown controller. Both intelligent system controllers are susceptible to EMI and power outages. They also need to be backed up by the conventional protection devices. The different protection strategies will operate as layers of protection each acting as a backup for the other layer of protection.

In both cases 5 and 6 the additions offer significant advantages, but where the safe operation of a large system is concerned, there cannot be compromised. This would mean that the additions are not to replace the conventional equipment, but rather to enhance the total system operation.

8.3 Future Work

The system additions has been tested at full power rating, but only in the protected environment of the laboratory. This is also the case for all the individual system controllers of the different subsystems. There are some future work to be done on the system.

- The additions to the practical system (The anti-parallel chopper and inverter and the high voltage dc dump) need to be installed and tested in a practical environment.
- Conventional protection devices need to be installed together with the intelligent system protection currently employed to protect the individual subsystems. The dual operation, as well as any interaction between the layers of protection need to be studied.
- The system controllers need to be expanded to operate on several substations and trains.
- The fuzzy controller can be expanded to monitor more parameters and a peak value algorithm can be implemented. Peak currents and short term power surges can be monitored and allowed by the fuzzy system controller. Time can be used as a parameter, if the controller is fast enough, and then implemented to optimise the systems operation.
- The fuzzy controller can be developed further to employ neural network operation. This would give the controller the ability to optimise itself over a length of time. The system could then "learn" how to give the best response to every error that occurs. Downtime could be minimised by utilising the system to its maximum as found by experience.
- An economic evaluation of the effect of the installation of the system additions need to be made and quantitative estimations made of the time it would take to cover the initial investment made with installation.

References

References

Journal Papers

[A1] Suzuki, T. "DC Power Supply System with Inverting Substations for Traction Systems using Regenerative Brakes", B Proc. IEE, Vol. 129, No. 1, Jan 1982, pp. 18 - 26.

[A2] Kneschke, T.A., "Simple Method for Determining of Substation Spacing for AC and DC Electrification systems", *IEEE Trans. on Ind. Appl.*, Vol. 22, July/Aug. 1986, pp. 763-780.

[A3] Forsythe, J.B., "Light Rail/Rapid Transit: New Approaches for the Evaluation of Energy Savings, Part II - On the Receptivity of a Transit System", *IEEE Trans. on Ind. Appl.*, Vol. 16, 1980, pp. 665-678.

[A4] Australian Standards Authority : "Disturbances in Mains Supply Network, AS2279", ASA, Part 1 and 2, Canberra, 1979.

[A5] British Standards Institution : "The Limitation of Disturbances in Electricity Supply Networks caused by Domestic and Similar Appliances Equipped with Electronic Devices", British Standard BS5406 (CENELEC 50.006), BSI, London, 1976.

[A6] Electricity Council : "Limits for Harmonics in the United Kingdom Electricity Supply System", Engineering Recommendation G5/3, EC, London, 1976.

[A7] Institute of Electrical and Electronic Engineers, "Guide for Harmonic Control and Reactive Compensation of Static Power Converters", IEEE, New York, 1981.

[A8] New Zealand Ministry, "Limitations of Harmonic Levels Notice", Wellington, 1981.

[A9] Koch, R, "Non-linear Loads in Utility Networks", Quality of Supply Conference, March 1995.

[A10] Enslin, J.H.R., Van Wyk, J.D., "Measurement and Compensation of Fictitious Power Under Non-sinusoidal Voltage and Current Conditions," *IEEE Trans. on Instrumentation and Measurement*, vol. 37, no. 3, September 1988, pp. 403-408.

[A11] Fujita, H., Akagi, H., "A Practical Approach to Harmonic Compensation in Power Systems - Series Connection of Passive and Active Filters," *IEEE Trans. on Ind. Appl.*, vol. 27, no. 6, November/December 1991, pp.1020-1025.

[A12] Marshall, D.A., "An Evaluation of the Real-Time Compensation of Fictitious Power in Electric Energy Networks," *IEEE Trans. on Power Delivery*, vol. 6, no. 4, October 1991, pp. 1774-1780.

[A13] Enslin, J.H.R., Van Wyk, J.D., "A new control philosophy for power electronic converters as fictitious power compensators", *IEEE Trans. on Power Electr.*, vol. 5, January 1990, pp. 87 - 97.

[A14] Czarnecki, L.S., "Considerations of Reactive Power in Non-sinusoidal Situations", *IEEE Trans. on Instrum. Meas.*, Vol. 34, No. 3, September 1983, pp. 399-404.

[A15] Enslin, J.H.R., Van Wyk, J.D., Naudè, M., "Adaptive, Closed-Loop Control of Dynamic Power Filters as Fictitious Power Compensators", *IEEE Trans. on Ind. Electr.*, vol. 37, no.3, June 1990, pp. 203-211.

[A16] Akagi, H., Kanazawa, Y., Nabae, A., "Instantaneous Reactive Power Compensators Comprising Switching Devices without Energy Storage Components," *IEEE Trans. on Ind. Appl.*, vol. IA-20, no. 3, May/June 1984, pp. 625-630.

[A17] Sasaki, H., Machida, T., "A New Method to Eliminate AC Harmonic Currents by Magnetic Compensation - Considerations on Basic Design", *IEEE Trans. PAS*, vol. 90, no. 5, 1971, pp. 2009-2019.

[A18] Bhagwat, P.M., Stefanovic, V.R., "Generalised Structure of a Multilevel PWM inverter", *IEEE Trans. on Industry Appl.*, Vol. IA-19, No. 6, November/December 1983, pp. 1057-1069.

[A19] Keraluwala, M., Divan, D.M., "Delta Modulation Strategies for Resonant Link Inverters", *IEEE Trans. on Power Electron.*, Vol. 5, No. 2, 1990, pp. 220-228.

[A20] Mertens, A., "Performance Analysis of Three-Phase Inverters Controlled by Synchronous Delta-Modulation Systems", *IEEE Trans. on Industry Appl.*, Vol. 30, No. 4, July/August 1994, pp. 1016-1027.

[A21] Czarnecki, L.S., "Measurement of the Individual Harmonics and Reactive Power in Non-sinusoidal Systems", *IEEE Trans. on Instrum. Meas.*, Vol. 32, No. 2, June 1983, pp. 383-384.

[A22] Huang, S, Lian, R, "A Hybrid Fuzzy Logic and Neural Network Algorithm for Robot Motion Control", *IEEE Trans. on Ind. Electron.*, Vol. 44, No. 3, June 1997, pp. 408-417.

[A23] Shaw, I.S., Mörs, W., Van Wyk, J.D., "Embedded Fuzzy Controller for a DC Locomotive", *SAIEE Transactions*, June 1996, pp. 65-76.

[A24] Lotfi, Z. "Fuzzy Sets", *IEEE Trans. on Information and Con[illegible]l* Vol. 8, pp 338-353, 1965.

Conference Papers

[B1] Randewijk, P-J., Enslin, J.H.R., "Inverting DC Traction Substation with Active Filtering Incorporated," *PESC'95 Conf. Proceedings*, Atlanta, June 1995, pp.360-366.

[B2] Tuncay, R.N., Kaypmaz, A., Yilmaz, .M., Brown, P.J., "The Theoretical and Experimental Study of AC Harmonics of Twelve-Pulse Railway Rectifiers", *EPE 1993*, pp. 197-204.

[B3] Horn, A, Wilkinson, R.H., Enslin, J.H.R., "Evaluation of Converter Topologies for Improved Power Quality in dc Traction Substations", *In Proc. IEEE ISIE'96*, Poland, pp. 802-807, June 1996.

[B4] Meynard, T.A., Foch, H., "Multilevel Conversion: High Voltage Choppers and Voltage-Source Inverters", *in PESC'92 Conference Proc.*, Toledo, Spain, pp. 397-403.

[B5] Meynard, T.A., Foch, H., "Multilevel Converters and Derived Topologies for High Power Conversion", *in PESC'95 Conference Proc.*, 95CH35818, pp. 21-26.

[B6] Wilkinson, R.H., Putter, C., Randewijk, P-J., Enslin, J.H.R., "DC Side Harmonic Compensation in DC Traction Applications", *In Proc. Africon'96*, Stellenbosch, pp. 827 - 832.

[B7] Le Roux, A.D., Frosch, M., Enslin, J.H.R., "A High Performance 100 kVA Four Quadrant Drive Test-bed", *In Proc. Africon'96*, Stellenbosch, pp. 462 - 466.

[B8] Gyugi, L., Strycula, E.C., "Active AC Power Filters", *Proc. IEEE IAS Annual Meeting*, 1976, pp. 529.

[B9] Horn, A., Pittorino, L.A., Enslin, J.H.R., "Evaluation of Active Power Filter Control Algorithms under Non-Sinusoidal and Unbalanced Conditions", *ICHQP'96 Proc.*, October 1996.

[B10] Hochgraf, C., Lasseter, R., Divan, D., Lipo, T.A., "Comparison of Multilevel Inverters For Static VAR Compensation", *in IAS'94*, pp. 921-928.

[B11] Lorenz, R.D., Divan, D.M., "Dynamic Analysis and Experimental Evaluation of Delta Modulators for Field Orientated AC Machine Current Regulators", *IEEE- IAS Conf. Rec.*, 1987, pp. 197-201.

[B12] Baldo, N., Malesani, L., et. al., "Hybrid Active Filter for Parallel Harmonic Compensation," *EPE 1993*, pp. 133-138.

[B13] Trzynadlowski, A.M., "An Overview of Modern PWM Techniques for Three-Phase, Voltage-Controlled, Voltage-Source Inverters", *Proc ISIE'96*, 96TH8192, pp. 25-39.

[B14] Kirlin, R.L., Legowski, S.F., Trzynadlowski, A.M., "An Optimal Approach to Pulse Width Modulation in Power Inverters", *in PESC'95 Conference Proc.*, 95CH35818, pp. 313-318.

[B15] Senjyu, T, Molinas, M, Uezata, K, "Power system stabilising control for FACTS devices using an adaptive fuzzy controller", *In Proc. of IPEC'97*, Singapore, p178-183, May 1997.

Text Books and Dissertations

[T1] Arrillaga, J., Bradley, D.A., Bodger, P.S., *Power System Harmonics*, New York: John Wiley and Sons, 1985.

[T2] Randewijk, P-J. *Inverting DC Traction Substation with Active Power Filtering Incorporated*, M. Eng. Thesis, University of Stellenbosch, December 1995.

[T3] Steyn, B., *The Effect of Harmonics on Railway Signalling Systems*, Ph.D. Thesis, RAU, 1995.

[T4] Miller, T.J.E., *Reactive Power Control in Electric Systems*, New York: John Wiley and Sons, 1982.

[T5] Horn, A, *Control of High Power Active Filter for DC Traction Substation*, M. Eng. Thesis, University of Stellenbosch, December 1996.

[T6] Le Roux, A.D., *Development of a Dynamic Train Load Simulator using a Four Quadrant AC-AC 100 kVA Machine Drive*, M. Eng. Thesis, University of Stellenbosch, December 1997.

[T7] Williams, B.W., *Power Electronics, Devices, Drivers, Applications and Passive Components*, Second Edition, 1992, pp. 359-371.

[T8] Du Toit, J.A., *Design, Implementation and Evaluation of Current Regulators for High Power Converters*, M. Eng. Thesis, University of Stellenbosch, February 1996.

[T9] Putter, C., *Development of a Converter for High Voltage DC-link applications*, M. Eng. Thesis, University of Stellenbosch, December 1997.

[T10] Yager, R.R., and Zadeh, L.A., *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer Academic Publishers, 1991.

[T11] De Beer, L.S., *Determination of Power Components in Non-linear Electrical Networks*, Masters Thesis, RAU, 1990.

[T12] Sen, P.C., *Principles of Electric Machines and Power Electronics*, Singapore : John Wiley and Sons, 1989.

[T13] Mohan, N., Undeland T..M., Robbins, W.P., *Power Electronics: Converters, Applications and Design*, New York: John Wiley and Sons, 1989.

[T14] *EDR Software developers kit for Eagle Technology boards*, Ver 1.4, Eagle Technology, Capetown, February 1996.

[T15] *User manual for the PC30F and G Series A/D Boards*, Fifth Edition, Eagle Technology, Capetown, March 1996.

Miscellaneous

[M1] Internal Spoornet Report EP.002, Issue 1, Sections 13, 15, 16, 31, Spoornet SA, August 1997.

[M2] Suzuki, T., "Traction substations in an energy-saving regenerative braking system", *Mitsubishi Electrical ADVANCE*, June 1979, pp. 1 - 4.

[M3] "Rationalised User Specification - Electricity Supply - Quality of Supply", *NRS048*, First Edition, Parts 1, 2, 3, 4, South African Bureau of Standards, 1996.

[M4] Marshall, D.A., Lewis, C.W.A., Du Toit, H.W., "Standard for maximum Voltage Distortion in Electrical Networks", *ESKOM Standard*, ESKASAA18, March 1994.

[M5] Topham, G.H., "Minimum Requirements for the Connection of Non-ESKOM Generating Plant to the ESKOM Distribution System", *ESKOM Guide*, ESKAGAAG2, July 1995.

[M6] IGBTMOD and Intellimod - Intelligent Power Modules Applications and Technical Data Book, Powerex, Pennsylvania, October 1994.

[M7] "Power Modules Designers Manual", International Rectifier, 1991, p. D349.

[M8] Beaud, E., Design and Development of a 4 kV, 50 A IGBT Solid-state DC Contactor, SPOORNET 9701, June 1997.

[M9] Randewijk, P-J, "Halfgeleierskakelaar vir die beheer van geregenereerde traksiedrywing", B.Eng. Thesis, University of Stellenbosch, December 1993.

[M10] King, P.J., Mamdani, E.H., "The Application of Fuzzy Control Systems to Industrial Processes", *Automatica*, vol. 13, pp. 235-241, 1977.

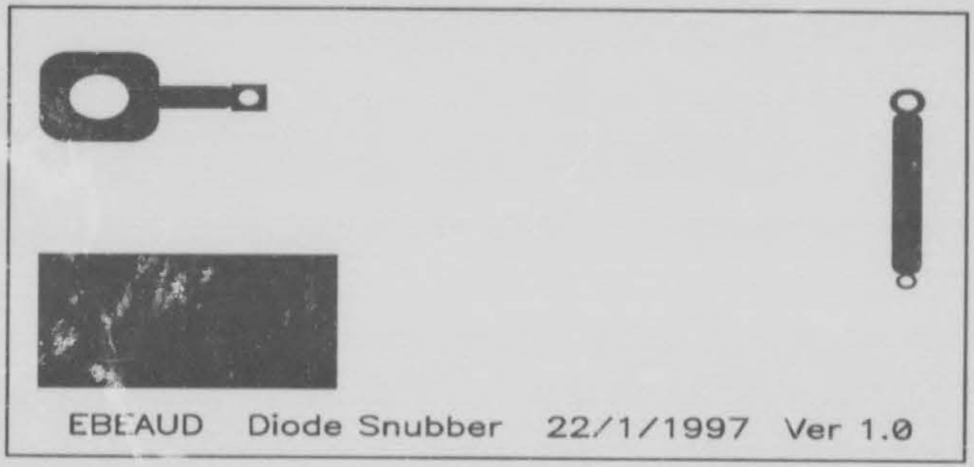
[M11] Pfitzinger, B., "A Fuzzy Logic Primer", pp. 11-14, *Quantum*, May 1997.

Appendix A

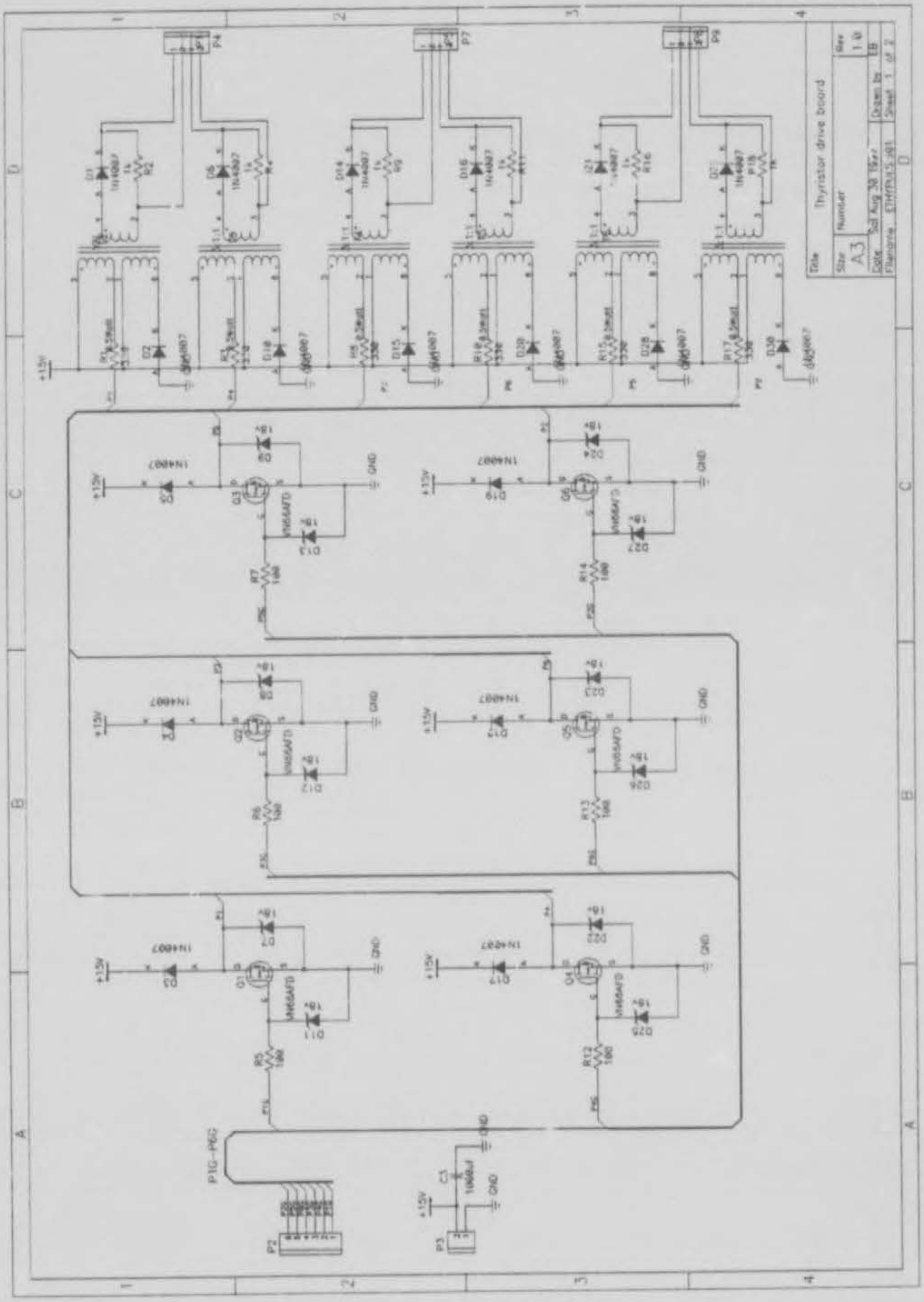
PC Boards

Appendix A1 - Diode Rectifier RC Snubber PCB (Not to Scale)

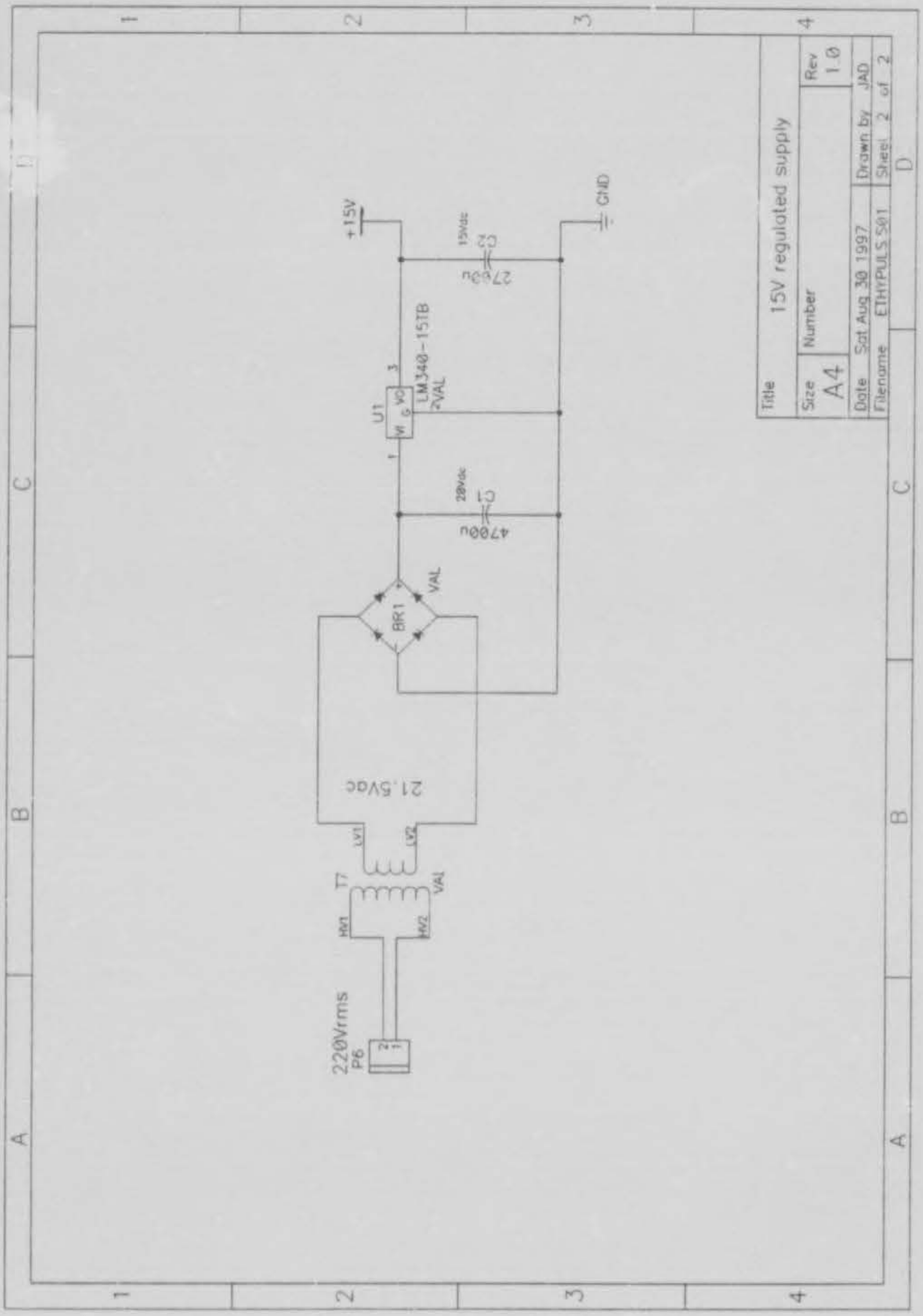
Diode Snubber Board
Component Side
EBEAUD
VER 1.0
Quantity : 24
Scale 2:1



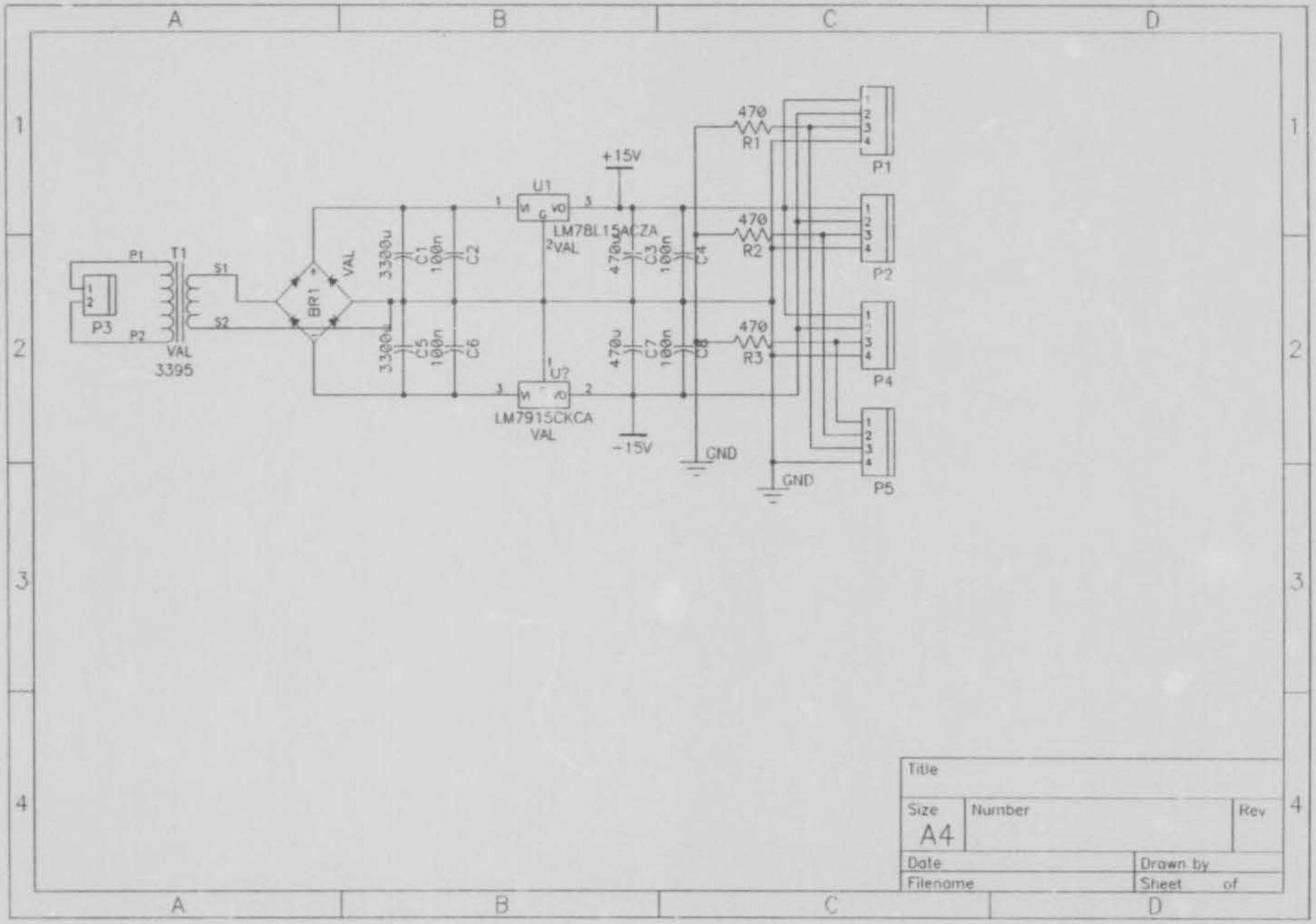
Appendix A2 - Three Phase Soft-starter Drive Board - Part I



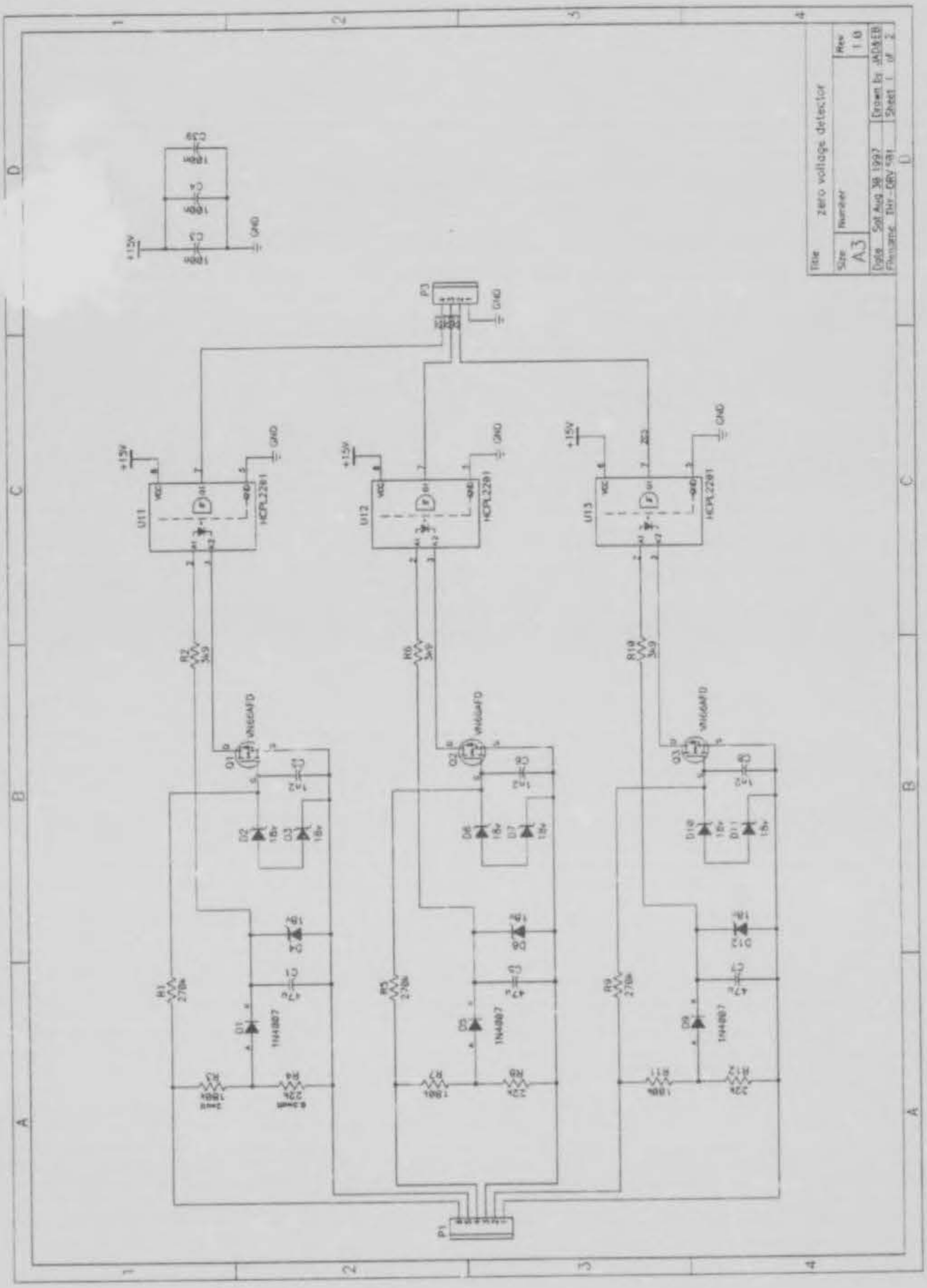
Appendix A2 - Three Phase Soft-starter Drive Board - Part II



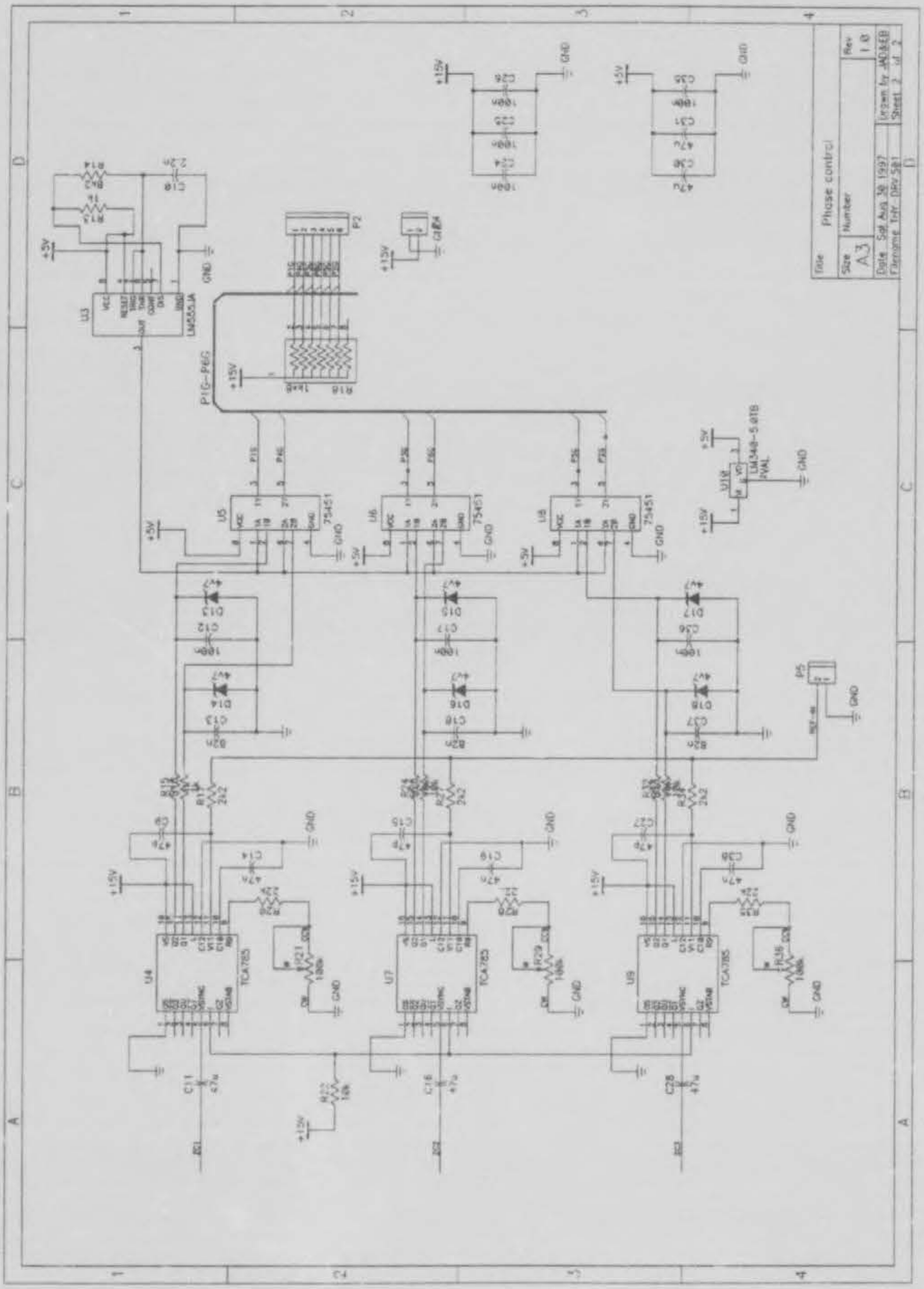
Appendix A3 - Current Measurement Driver Board



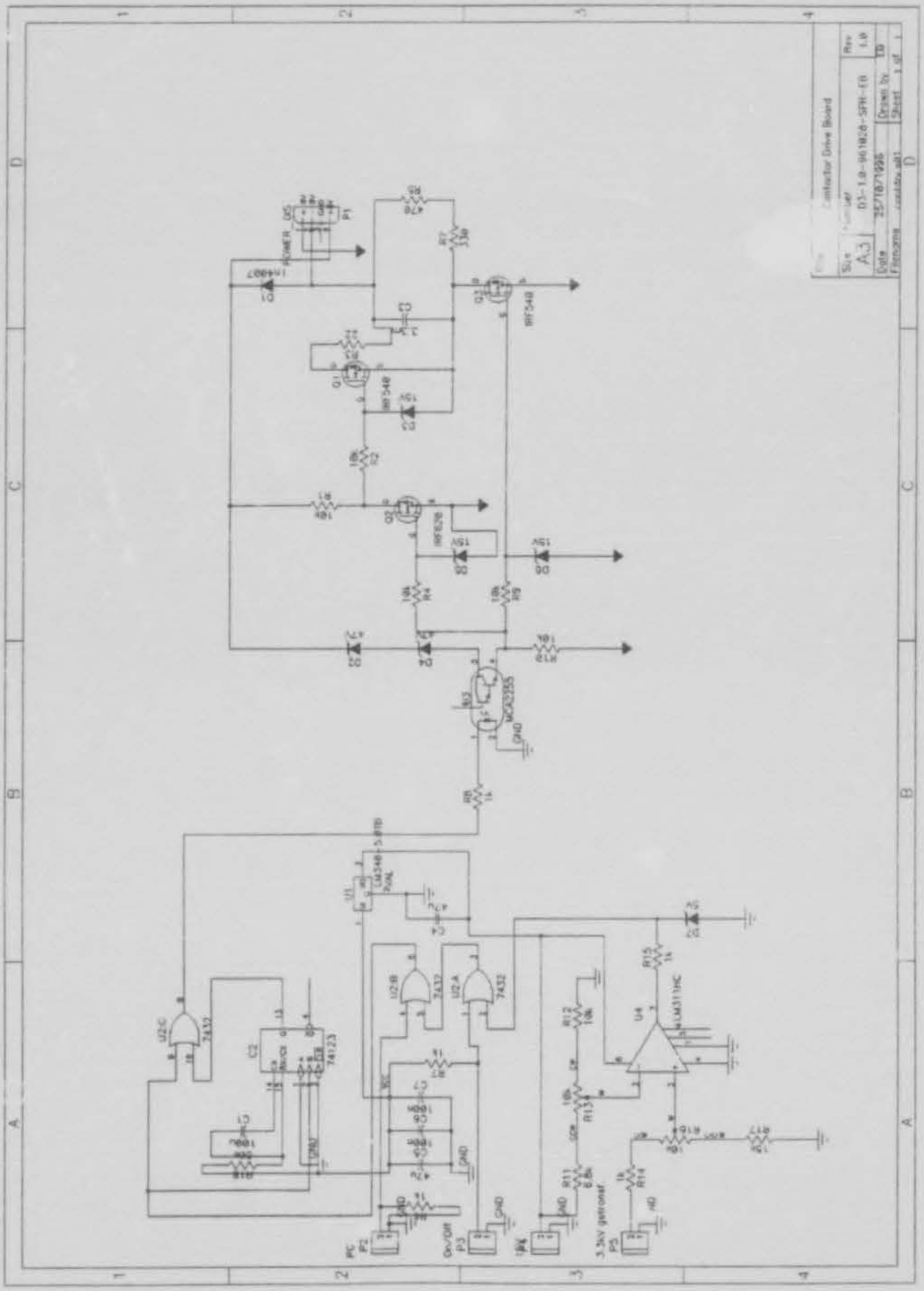
Appendix A4 - Soft-Starter Firing Angle Control Board - Part I



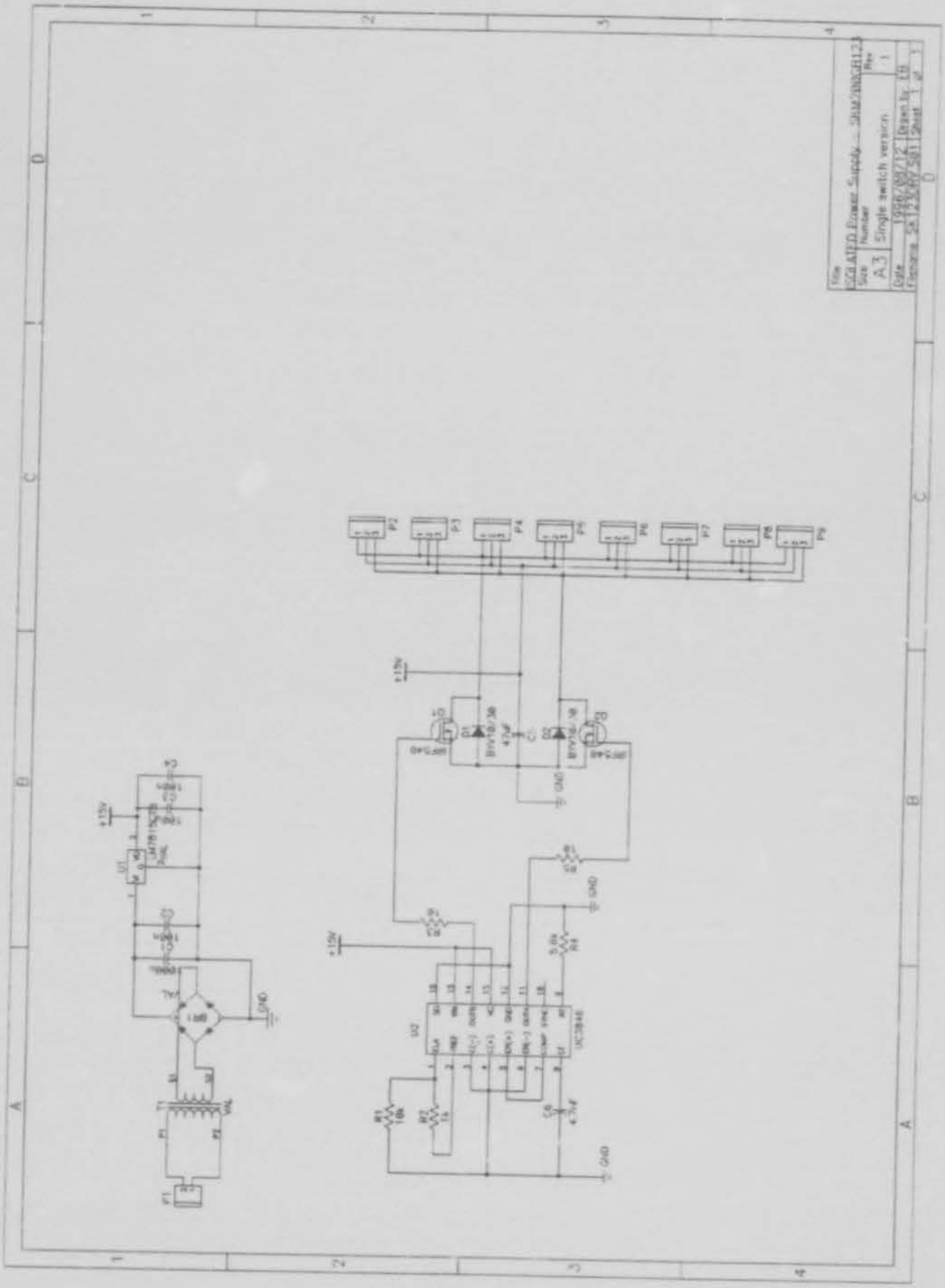
Appendix A4 - Soft-Starter Firing Angle Control Board - Part I



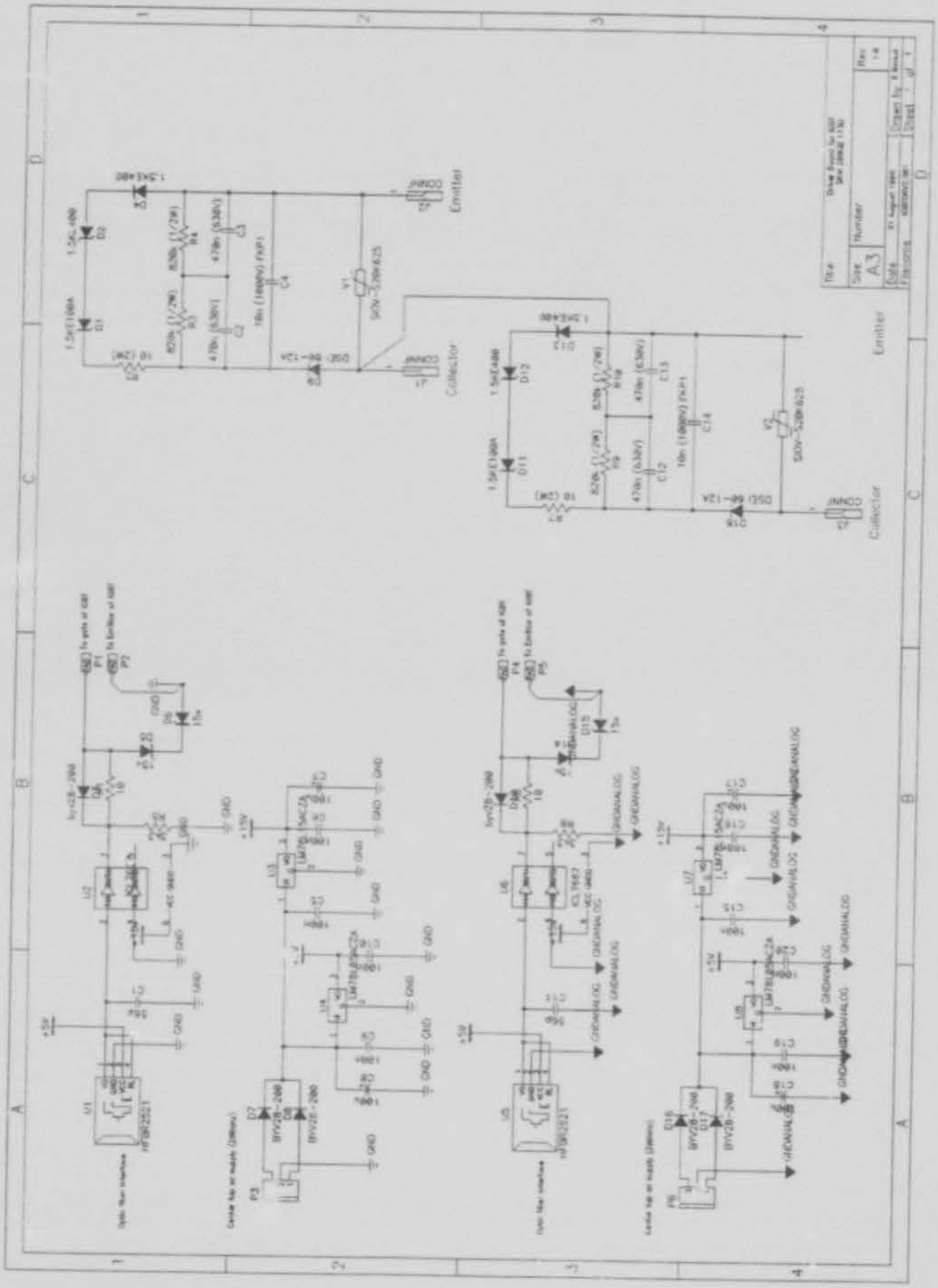
Appendix A5 - DC Contactor Drive Board



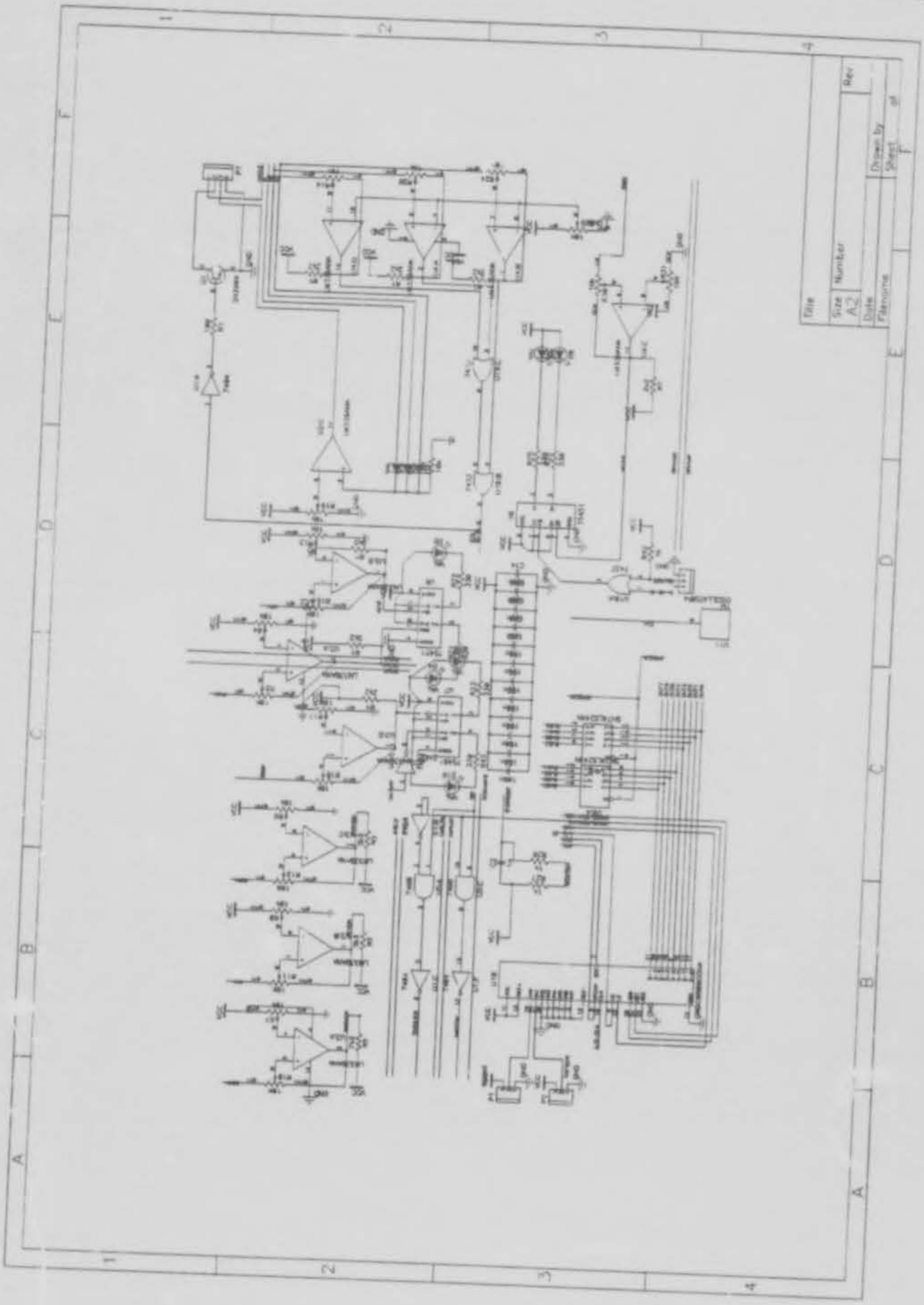
Appendix A6 - DC Dump Isolated Powar Supply Driver Board



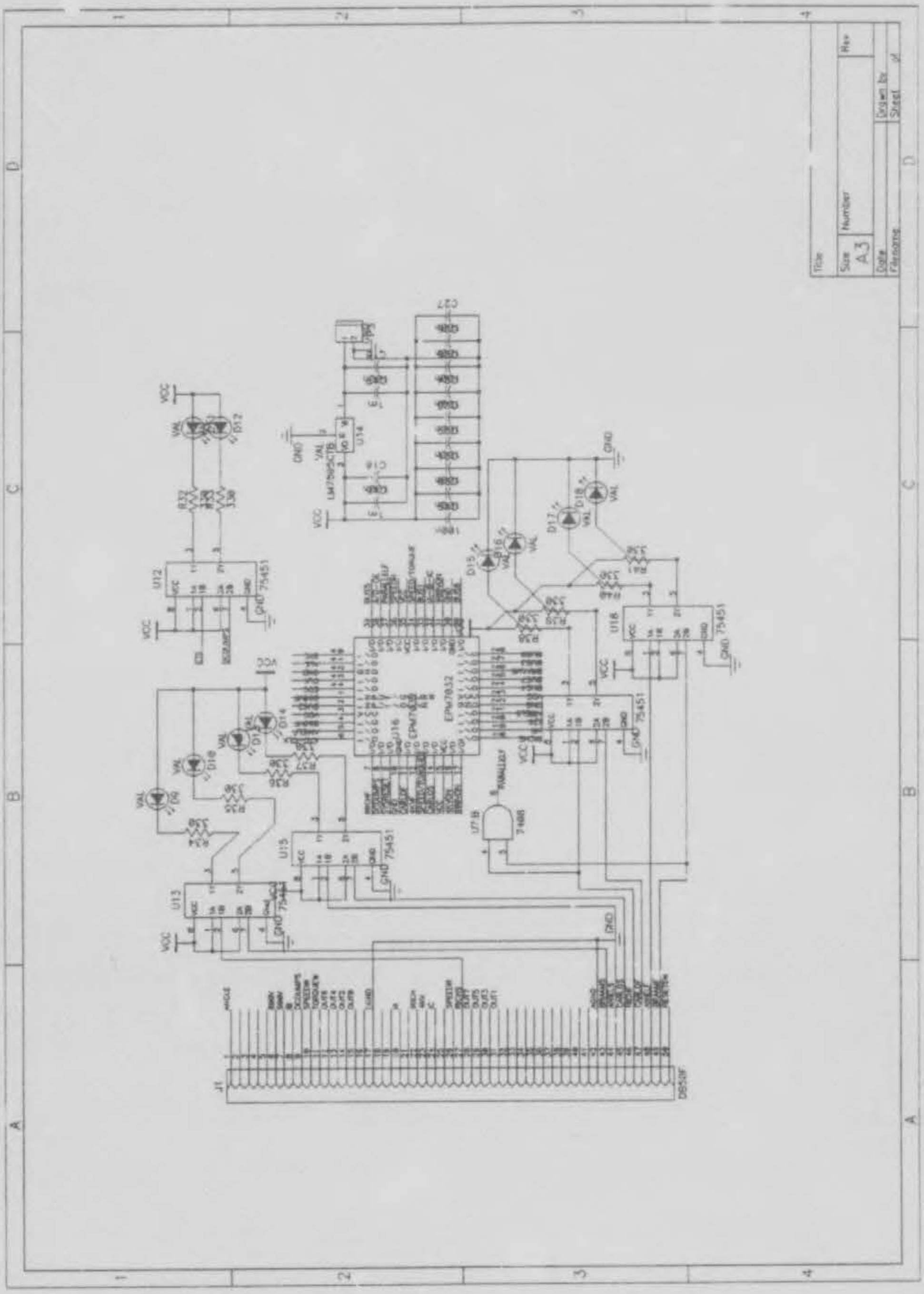
Appendix A7 - DC Dump Module Drive Boards and Snubbers



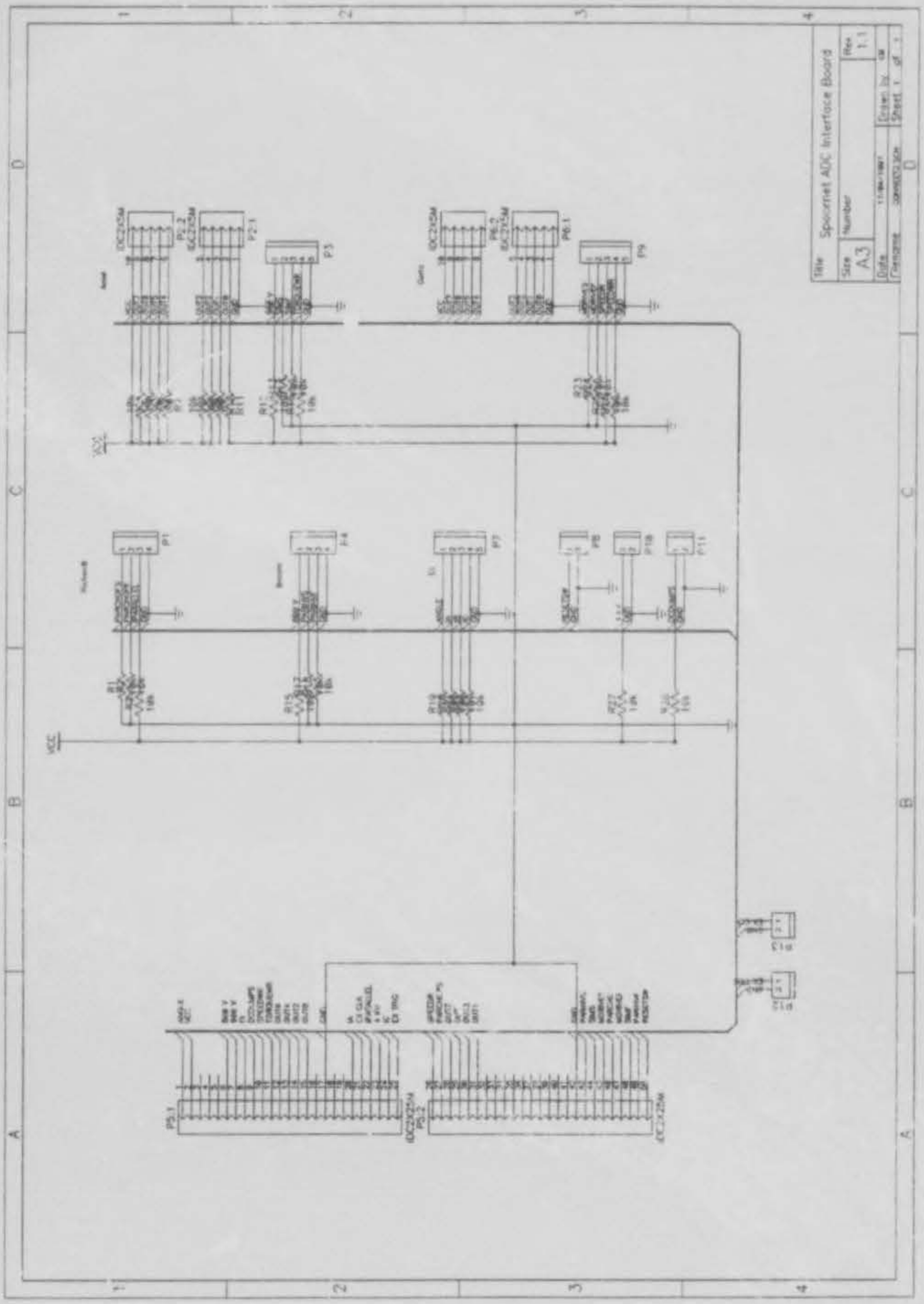
Appendix A8 - EPLD Controller Board - Part I



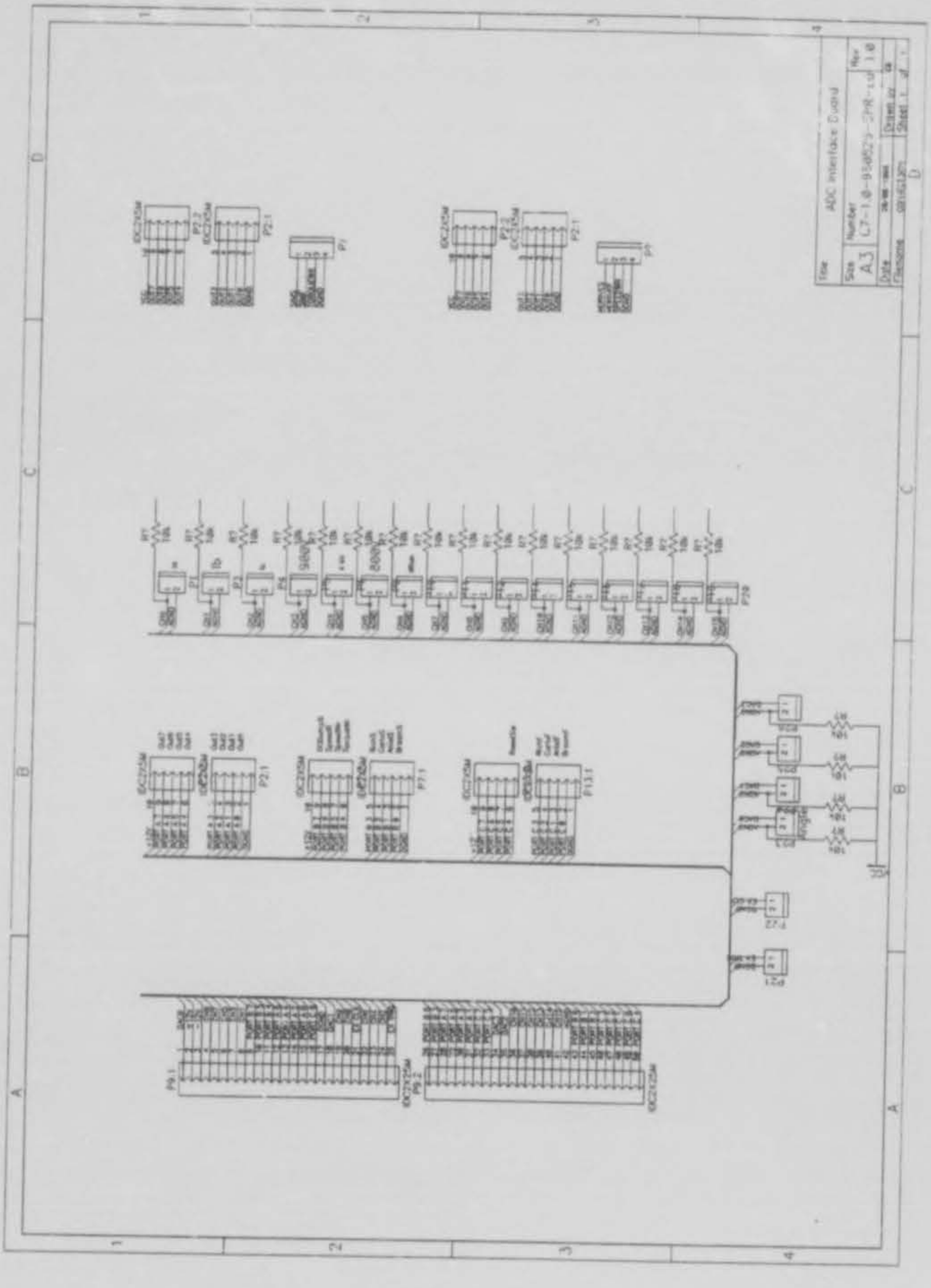
Appendix A8 - EPLD Controller Board - Part II



Appendix A8 - Scale System - EPLD Controller Connection Board



Appendix A10 - PC30GA A/D Converter Connection Board



Appendix B

RC Snubber Design for Freewheel Diodes of the DC Dump

Appendix B - RC Snubber design for freewheel diodes of DC Dump

The RC snubber for the diodes have to be designed for the worst case scenario to ensure safe operation. This condition is hard to define because the inductance in the path of the diodes is not known. It can be estimated from experience with leakage inductance or it could be measured using high accuracy measurement instruments. In the design of the dc dump neither the measurement instruments nor the time was available to take an exact measurement of the leakage inductance. The inductance was estimated not to be greater than 1 μH. The maximum reverse recovery current could be calculated using equation 1.

$$I_{RR} = 2.8\times10^{-6}\cdot BV_{BD}\sqrt{I_F \, {}^{dI_R}\!/_{dt}} \qquad \text{[T13]} \ldots\ldots 1$$

The breakdown voltage of the diode (SKN140F17) is given in the datasheets as 1.7 kV. The worst case forward current is 100 A and the rate of change of current during switch-off is limited by the leakage inductance. The datasheets also give a graph for the peak reverse recovery current as a function of the rate of change in the diode current. With the inductance taken as 1 μH it was not necessary to calculate the current, but it was read as 115 A. The RC snubber is studied in detail in [T13] and optimisation curves are drawn to choose the best value for the RC snubber (Figure 19-16 in [T13]).

First an optimal value for the snubber capacitor has to be chosen. The smaller the capacitor the less power needs to be dissipated in the snubber resistor. The bigger the capacitor the less the voltage overshoot at snap-off. A baseline capacitance corresponding to $1/\sqrt{2}$ overshoot can be calculated using equation 2.

$$C_{Base} = L_\sigma\cdot\left[\frac{I_{RR}}{V_d}\right]^2 \ldots\ldots 2$$

V_d is the dc bus voltage. The base line capacitance was calculated to be 4.6 nF. The corresponding base line resistance is given by

$$R_{base} = \frac{V_d}{I_{RR}} \ldots\ldots 3$$

The base line resistor is calculated as 14.8 Ω. A small drop in the capacitor value produces a significant drop in power consumption, but only a small rise in the voltage. A final value of 80 % of the baseline value (3.7 nF) was chosen which corresponds to an overshoot of 80 %. The optimal resistance is then 1.3 times the baseline resistance. The snubber resistance is then 19.24 Ω.

The energy dissipation in the resistor in one cycle is the energy stored in the leakage inductance plus that stored in the capacitor. This is given in equation 4

$$W_R = \frac{1}{2} L_\sigma I_{ss}^2 + \frac{1}{2} C_s V_d^2 \quad \text{.....4}$$

With an inductance of 1 μH and a capacitance of 3.7 nF the energy dissipated in one cycle is 11.9 mJ. With a switching frequency of 1 kHz the means that the resistor will have to dissipate 11.9 W.

The capacitors in the right range that was available, was 10 nF, 1500 V and 4.7 nF, 1000 V snubber capacitors. Neither one capacitor has a high enough voltage rating but the series combination of them sees a large enough voltage rating with a total capacitance of 3.2 nF. This is very close to the designed value of 3.7 nF. The voltage will not share equally between the capacitors, but the bigger capacitance will take the smaller voltage (C = QV). The unbalance is small enough that neither capacitor will be damaged.

The only resistors that were available were 10Ω , 10 W resistors. These are a lot smaller than optimal , but close enough to work.

Appendix C

PI Controller Design for the DC Dump

Appendix C - PI Controller design for the DC Dump

In classical control theory a system controller is designed to have the system respond within specified limits. First the system is characterised and then the controller is designed to control the system such that the specifications are met. In the prototype system this line of design has certain problems.

In an attempt to characterise the system, a state space approach was followed to get a transfer function for the dc dump. The duty cycle was used to take an average between the on-state and the off-state of the IGBT switch. The impedance of the rest of the system was taken to be purely resistive to simplify the design. The relation between the dc bus voltage and the duty cycle could be written as follows

$$V_{DC} = V_{NL} - DV_{NL}\left[\frac{(V_{NL} - V_{FL})R}{V_{FL}Ls + V_{NL}R}\right] \qquad (1$$

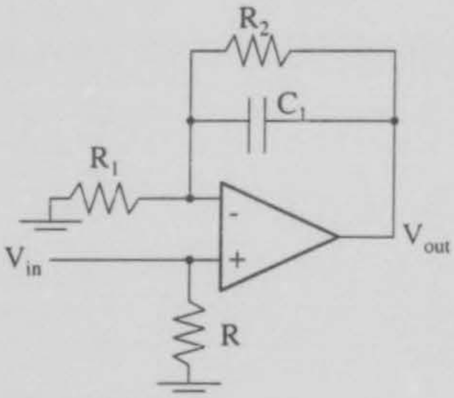
It can be seen that the duty cycle does not control the dc bus voltage directly, but rather the fall in dc bus voltage (V_{Delta}). The transfer function for the dc dump can now be written as

$$\frac{V_{Delta}}{D} = V_{NL}\left[\frac{(V_{NL} - V_{FL})R}{V_{FL}Ls + V_{NL}R}\right] \qquad (2$$

If a PI controller is used to control the dc dump, it will add a zero and a pole at zero to the open loop transfer function. The dc dump has a pole at $-V_{NL}R/V_{FL}L$ and a zero at negative infinity. The PI controller was designed to cancel the pole of the dc dump. This keeps the system of the first order. The loop gain was chosen to be as high as the noise ratio allowed, to give the fastest system response.

The proportional term was simply chosen as unity to ensure that the dc bus voltage would never be able to rise unchecked. The maximum rated voltage for the dc bus is 5000 V which corresponds to 5 V in the control circuitry. In the case of such a sudden rise the controller would immediately react with the maximum measure of compensation. The integration gain was calculated to be 1670 to cancel the pole of the dc dump. The loop gain was started at unity.

The PI controller was implemented using a single operational amplifier. The circuit is shown in Figure 1.



Appendix E - Figure 1 : PI Controller for the dc dump

The transfer function of the above circuit is given in equation 1.

$$\frac{V_{Out}}{V_{In}} = \frac{1}{R_1\left(C_1 s + \frac{1}{R_2}\right)} + 1 \qquad (1)$$

If $1/R_2$ is chosen to be a lot smaller than C_2 then the transfer function reduces to a unity proportional part and an integrator with a gain of $1/R_1C_1$. R_2 was chosen as 10 MΩ, R_1 as 6.8 kΩ and C_1 as 82 nF. These values are calculated ignoring the transient response of the rest of the system. The gain was also chosen according to a worst case full step in error voltage. Neither of these assumptions are necessarily close to the truth and this should be taken into account in final adjustments made to the controller.

Appendix D

Delphi System Control Program

Appendix D1 - Delphi system control program - controller

```
unit Fuzzy;
{E. Beaud  : Spoornet system controller employing shutdown
and fuzzy protection strategies}

interface

uses {Libraries Used}
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Spin, Outline, DirOutln, Grids, Calendar, ExtCtrls, Buttons, EDR32,
  Gauges, Mask;

type {Type Definitions}
  TSystemController = class(TForm)
    SystemStati: TGroupBox;     {Screen Objects}
    SoftStarterStatus: TCheckBox;
    DCDriveStatus: TCheckBox;
    ParChopperStatus: TCheckBox;
    ParInverterStatus: TCheckBox;
    AcMotorDriveStatus: TCheckBox;
    SystemOnOffButton: TBitBtn;
    SystemErrors: TGroupBox;
    SysParameters: TGroupBox;
    DCMotorDriveError: TCheckBox;
    ParChopperError: TCheckBox;
    ParInverterError: TCheckBox;
    ACMotorDriveError: TCheckBox;
    Bus4kVOn: TCheckBox;
    ParBusOn: TCheckBox;
    AC_ACBusOn: TCheckBox;
    ParCurrentOn: TCheckBox;
    SystemCurrentOn: TCheckBox;
    DCDumpStatus: TCheckBox;
    Bus4kVError: TCheckBox;
    ParBusError: TCheckBox;
    AC_ACBusError: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    SYSONLED: TImage;
    SYSOFFLED: TImage;
    ParCurrentError: TCheckBox;
    SystemCurrentError: TCheckBox;
    SystemStartTimer: TTimer;
    Label3: TLabel;
```

```
SetUpButton: TButton;
SystemRunTimer: TTimer;
SystemStopTimer: TTimer;
ParFault: TTimer;
AC_ACFault: TTimer;
Running: TCheckBox;
DcDumpTimer: TTimer;
FaultReader: TTimer;
StatusWriter: TTimer;
DataWriter: TTimer;
DataReader: TTimer;
Label14: TLabel;
Label15: TLabel;
Helper: TTimer;
Fuzzification: TTimer;
GroupBox1: TGroupBox;
Gauge1: TGauge;
Gauge2: TGauge;
Gauge3: TGauge;
Gauge4: TGauge;
Gauge5: TGauge;
Gauge6: TGauge;
Gauge7: TGauge;
Gauge8: TGauge;
Gauge9: TGauge;
Gauge10: TGauge;
Gauge11: TGauge;
Gauge12: TGauge;
Gauge13: TGauge;
Gauge14: TGauge;
Gauge19: TGauge;
Gauge20: TGauge;
Label4: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label8: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label33: TLabel;
```

```
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
Label37: TLabel;
Label38: TLabel;
Medium: TLabel;
Label39: TLabel;
GroupBox2: TGroupBox;
Gauge17: TGauge;
Label17: TLabel;
Gauge15: TGauge;
Label5: TLabel;
Label16: TLabel;
Gauge16: TGauge;
Label18: TLabel;
Gauge18: TGauge;
GroupBox3: TGroupBox;
TorqueRef: TLabel;
TorqueRefGauge: TGauge;
Speed: TGauge;
Label6: TLabel;
Label7: TLabel;
Speedref: TLabel;
SpeedRefGauge: TGauge;
O: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
TorqueStatus: TCheckBox;
SpeedStatus: TCheckBox;
SpeedRequired: TMaskEdit;
TorqueRequired: TMaskEdit;
TorqueSign: TListBox;
Start: TBitBtn;
Stop: TBitBtn;
Label9: TLabel;
Label40: TLabel;
Label41: TLabel;
Label42: TLabel;
Label43: TLabel;
Label44: TLabel;
Label45: TLabel;
Label46: TLabel;
Label47: TLabel;
Label48: TLabel;
Label49: TLabel;
Label50: TLabel;
procedure SystemOnOffButtonClick(Sender: TObject); {Procedures}
```

```
    procedure SystemStartTimerTimer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SetUpButtonClick(Sender: TObject);
    procedure StartClick(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure SystemRunTimerTimer(Sender: TObject);
    procedure ParFaultTimer(Sender: TObject);
    procedure AC_ACFaultTimer(Sender: TObject);
    procedure SystemStopTimerTimer(Sender: TObject);
    procedure StopClick(Sender: TObject);
    procedure DcDumpTimerTimer(Sender: TObject);
    procedure StatusWriterTimer(Sender: TObject);
    procedure DataWriterTimer(Sender: TObject);
    procedure FaultReaderTimer(Sender: TObject);
    procedure DataReaderTimer(Sender: TObject);
    procedure FuzzificationTimer(Sender: TObject);
    procedure HelperTimer(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  {System Variables}
  SystemController: TSystemController;
  bh, bt : Integer;
  baseaddr : Integer;   { Board Specs }

  {Fuzzy Output Variables}
  Angle      : Real;    { Firing Angle for Soft-Starter }
  DeltaAngle : Real;    { Change in Firing Angle for Soft-Starter }
  DeltaSpeed : Real;    { Change in Speed Reference Signal}
  Torque     : Real;    { Torque Reference Signal}

  {Fuzzy Help Variables}
  CurrentAngle : Real;  { Internal Firing Angle variable}
  DeltaScreen : Real;   { Change in Speed Required by the user}
  ScreenTorque : Real;  { Torque Required by the user}
  TorqueProgressSize : Integer; { Current Torque Reference}

  {Results of rule evaluations - rule base}
  Alpha1, Alpha2, Alpha3, Alpha4, Alpha5, Alpha6, Alpha7, Alpha8 : Real;
  Alpha9, Alpha10, Alpha11, Alpha12, Alpha13, Alpha14, Alpha15 : Real;
  AlphaACCurrentsHigh : Real; { Combined fuzzification result
                                of three ac currents}

  {Fuzzy Input Variables}
```

```
 Ia, Ib, Ic       : integer;  { Phase Currents }
 V900, V800, V4kV : integer; { DC Bus Voltages }
 Ipar             : integer; { Parallel Current }
 CurrentSpeed     : Real; { Current Motor Speed }

 {System Outputs}
 SpeedOut : Real;

const
 NumChans  = 7;             { Number of Channels in Sample List - Inputs }
 TimerCount = 5000;          { Timer Clock Frequency}
 ADClockmilliHz = 10000000;  { AD Clock Frequency in milliHertz}
 Output = 0; {Port Direction Definitions}
 Input = 1;
 PortA = 0;  {Port Number Definitions}
 PortB = 1;
 PortC = 2;

 {System Parameter Limits}
 {See that these are outside the fuzzy limits}

 {AC Input Currents}
 CurrentOnLimit    = 50000;
 CurrentErrorLimit = 4500000;

 {900V Bus Voltage}
 V900BusOnLimit    = 3000000;
 V900BusErrorLimit = 4500000;

 {4 kV Bus Voltage}
 V4kVBusOnLimit    = 2900000;
 V4kVBusErrorLimit = 4800000;

 {800V Bus Voltage}
 V800BusOnLimit    = 2500000;
 V800BusErrorLimit = 4500000;

 {Parallel Bus Current}
 ParallelCurrentOnLimit    = 50000;
 ParallelCurrentErrorLimit = 4500000;

 {Speed}
 RunningSpeed = 100;
 MaxRPM       = 1500; { Maximum motor speed}

 {Fuzzification Limits}
 {See that these are inside the on-off limits}

 {INPUTS}
```

```
{AC Input Current}
IacMed   = 2000000;
IacHigh   = 3000000;
IacVeryHigh = 4000000;

{4kV Bus Voltage}
Bus4kVLowMin     = 1000000;
Bus4kVLowPlus    = 2000000;
Bus4kVRight      = 3500000;
Bus4kVRightDelta = 200000;
Bus4kVDelta      = 500000; {Limit Delta according to Bus4kVVeryHigh}
Bus4kVHigh       = Bus4kVRight + Bus4kVDelta;
Bus4kVVeryHigh   = 4500000;

{Synchronous Drive Bus Voltage}
SynchroBusRight  = 3500000;
SynchroBusRightDelta = 200000;
SynchroBusDelta     = 500000;

{Parallel Bus Voltage}
ParallelBusRight = 3500000;
ParallelBusRightDelta = 200000;
ParallelBusDelta = 500000;

{Parallel Bus Current}
ParallelCurrentMedium = 3000000
ParallelCurrentHigh   = 3500000;

{Speed}
SpeedOKLimit = 1000;
SpeedHighLimit = 1200;

{OUTPUTS}
{Angle}
AngleZero = 0;
AngleMed = 90;
AngleHigh = 180;

{DeltaAngle}
DeltaAngleHighNeg = -20;
DeltaAngleMedNeg  = -5;
DeltaAngleZero    = 0;
DeltaAngleMedPos  = 5;
DeltaAngleHighPos = 20;

{DeltaSpeed - The difference between real speed and Speedreference}
DeltaSpeedLow      = 0;
DeltaSpeedMedNeg   = -200;
DeltaSpeedLowRange = 100;
```

```
  DeltaSpeedMedRange = 1000;
  DeltaSpeedHighRange = 1500;

  {Torque}
  TorqueHighNeg = 0;
  TorqueMedNeg  = 63;
  TorqueLow     = 127;
  TorqueMedPos  = 190;


implementation

uses Setup;

{$R *.DFM}

{Membership Functions for Inputs}

{Is ac input current high?}
function HighCurrent (Iin: integer): real;
begin
  if Iin < IacMed then HighCurrent := 0
  else if (Iin > IacMed) and (Iin < IacHigh) then
     HighCurrent := (Iin - IacMed)*100 div (IacHigh - IacMed)
  else HighCurrent := 100;
end;

{Is ac input current dramatically high?}
function VeryHighCurrent (Iin: integer): real;
begin
  if Iin < IacHigh then VeryHighCurrent := 0
  else if (Iin > IacHigh) and (Iin < IacVeryHigh) then
     VeryHighCurrent := (Iin - IacHigh)*100 div (IacVeryHigh - IacHigh)
  else VeryHighCurrent := 100;
end;

{Is synchronous dc bus voltage too low?}
function LowSynchroBus(Vin: integer): real;
begin
  if Vin > (SynchroBusRight - SynchroBusRightDelta) then LowSynchroBus := 0
  else if (Vin < (SynchroBusRight - SynchroBusRightDelta)) and
        (Vin > SynchroBusRight - SynchroBusDelta) then
        LowSynchroBus := ((SynchroBusRight - SynchroBusRightDelta) - Vin)*100
        / (SynchroBusDelta - SynchroBusRightDelta)
  else LowSynchroBus := 100;
end;

{Is synchronous dc bus voltage inside the right operating area?}
```

```
function RightSynchroBus(Vin: integer): real;
begin
  if abs(SynchroBusRight - Vin) > SynchroBusDelta then RightSynchroBus := 0
  else if Vin <= (SynchroBusRight - SynchroBusRightDelta) then
        RightSynchroBus := (Vin - (SynchroBusRight - SynchroBusDelta))*100
                        / (SynchroBusDelta - SynchroBusRightDelta)
  else if Vin >= (SynchroBusRight + SynchroBusRightDelta) then
        RightSynchroBus := ((SynchroBusRight + SynchroBusDelta) - Vin)*100
                        / (SynchroBusDelta - SynchroBusRightDelta)
  else RightSynchroBus := 100;
end;

{Is synchronous dc bus voltage too high?}
function HighSynchroBus(Vin: integer): real;
begin
  if Vin < (SynchroBusRight + SynchroBusRightDelta) then HighSynchroBus := 0
  else if (Vin > (SynchroBusRight + SynchroBusRightDelta)) and
        (Vin < SynchroBusRight + SynchroBusDelta) then
        HighSynchroBus := (Vin - (SynchroBusRight + SynchroBusRightDelta))*100
        / (SynchroBusDelta - SynchroBusRightDelta)
  else HighSynchroBus := 100;
end;

{Is parallel dc bus voltage inside right operating area?}
function RightparallelBus(Vin: integer): real;
begin
  if abs(parallelBusRight - Vin) > ParallelBusDelta then RightparallelBus := 0
  else if Vin <= (ParallelBusRight - parallelBusRightDelta) then
        RightparallelBus := (Vin - (ParallelBusRight - ParallelBusDelta))*100
                        / (ParallelBusDelta - ParallelBusRightDelta)
  else if Vin >= (ParallelBusRight + ParallelBusRightDelta) then
        RightparallelBus := ((ParallelBusRight + ParallelBusDelta) - Vin)*100
                        / (ParallelBusDelta - ParallelBusRightDelta)
  else RightparallelBus := 100;
end;

{Is parallel dc bus voltage too high?}
function HighParallelBus(Vin: integer): real;
begin
  if Vin < (ParallelBusRight + ParallelBusRightDelta) then HighParallelBus := 0
  else if (Vin > (ParallelBusRight + ParallelBusRightDelta)) and
        (Vin < ParallelBusRight + ParallelBusDelta) then
        HighParallelBus := (Vin - (ParallelBusRight + ParallelBusRightDelta))*100
        / (ParallelBusDelta - ParallelBusRightDelta)
  else HighParallelBus := 100;
end;

{Is parallel dc current too high?}
function HighParallelCurrent(Iin: integer): real;
```

```
begin
  if Iin < ParallelCurrentMedium then HighParallelCurrent := 0
  else if (Iin > ParallelCurrentMedium) and (Iin < ParallelCurrentHigh) then
      HighParallelCurrent := (Iin - ParallelCurrentMedium)*100 /
                        (ParallelCurrentHigh - ParallelCurrentMedium)
  else HighParallelCurrent := 100;
end;

{Is main dc bus voltage too high?}
function Low4kVBus(Vin: integer): real;
begin
  if Vin > Bus4kVLowPlus then Low4kVBus := 0
  else if (Vin < Bus4kVLowPlus) and (Vin > Bus4kVLowMin) then
      Low4kVBus := (Bus4kVLowPlus - Vin)*100 / (Bus4kVLowPlus - Bus4kVLowMin)
  else Low4kVBus := 100;
end;

{Is main dc bus voltage a little low?}
function Medium4kVBus(Vin: integer): real;
begin
  if (Vin < Bus4kVLowMin) OR (Vin > (Bus4kVRight - Bus4kVRightDelta))
    then Medium4kVBus := 0
  else if (Vin > Bus4kVLowMin) and (Vin < Bus4kVLowPlus) then
      Medium4kVBus := (Vin - Bus4kVLowMin)*100 / (Bus4kVLowPlus - Bus4kVLowMin)
  else if (Vin > (Bus4kVRight - Bus4kVDelta)) and
        (Vin < (Bus4kVRight - Bus4kVRightDelta)) then
      Medium4kVBus := (Bus4kVRight - Bus4kVRightDelta - Vin)*100 /
      (Bus4kVDelta - Bus4kVRightDelta)
  else Medium4kVBus := 100;
end;

{Is main dc bus voltage inside the right operating area?}
function Right4kVBus(Vin: integer): real;
begin
  if abs(Bus4kVRight - Vin) > Bus4kVDelta then Right4kVBus := 0
  else if Vin <= (Bus4kVRight - Bus4kVRightDelta) then
        Right4kVBus := (Vin - (Bus4kVRight - Bus4kVDelta))*100
                    / (Bus4kVDelta - Bus4kVRightDelta)
  else if Vin >= (Bus4kVRight + Bus4kVRightDelta) then
        Right4kVBus := ((Bus4kVRight + Bus4kVDelta) - Vin)*100
                    / (Bus4kVDelta - Bus4kVRightDelta)
  else Right4kVBus := 100;
end;

{Is main dc bus voltage too high?}
function High4kVBus(Vin: integer): real;
begin
  if Vin < (Bus4kVRight + Bus4kVRightDelta) then High4kVBus := 0
  else if (Vin > (Bus4kVRight + Bus4kVRightDelta)) and
```

```
        (Vin < (Bus4kVRight + Bus4kVDelta)) then
        High4kVBus := (Vin - (Bus4kVRight + Bus4kVRightDelta))*100 /
                (Bus4kVDelta - Bus4kVRightDelta)
  else High4kVBus := 100;
 end;

 {Is the rotor speed too high?}
 function HighSpeed(Vin: integer): real;
 begin
  if Vin < SpeedOKLimit then HighSpeed := 0
  else if (Vin > SpeedOKLimit) and (Vin < SpeedHighLimit) then
     HighSpeed := (Vin - SpeedOKLimit)*100 /
             (SpeedHighLimit - SpeedOKLimit)
  else HighSpeed := 100;
 end;


{Membership Functions for Control Outputs}
{This is for Centre of Gravity defuzzification method}

{Start of visual interface implementation}
procedure TSystemController.SystemOnOffButtonClick(Sender: TObject);
begin
 SysOffLed.Visible:= Not (SysOffLed.Visible);
 SysOnLed.Visible := Not (SysOffLed.Visible);
 IF SysOnLed.Visible THEN
  Begin
   Start.Visible := True;
   Stop.Visible := True;
   SystemOnOffButton.Caption := 'System On';
  End
 Else
  Begin
   Start.Visible := False;
   Stop.Visible := False;
   SystemOnOffButton.Caption := 'System Off';
   DCDriveStatus.Checked := False;  {Setup system stati at startup}
   ParChopperStatus.Checked := False;
   ParInverterStatus.Checked := False;
   ACMotorDriveStatus.Checked := False;
   SoftStarterStatus.Checked := False;
   DCDumpStatus.Checked := True;

   AC_ACFault.Enabled := False;      {Setup system fault signal}
   SystemStopTimer.Enabled := False; {stati at startup}
   ParFault.Enabled := False;
   SystemRunTimer.Enabled := False;
   SystemStartTimer.Enabled := False;
 End;
```

```
end;

procedure TSystemController.FormCreate(Sender: TObject);
var i :smallint;
begin
  bh:= EDR_AllocBoardHandle;  {Allocate a board handle to the A/D board}
  SystemController.Tag := bh;
  if bh=0 then  {Handle the case where there is no board}
   begin
     MessageDlg('Unable to allocate board handle',mtError,[mbOk],0);
     halt;
   end;
  baseaddr:=$280;  {Set board address}
  EDR_InitBoard(bh,baseaddr); {Initialise board}
  EDR_GetBoardType(bh,bt);   {Identify board}
  if (bt<PC30GA) or (bt>PC30GS4) then
   begin  {Handle case where there is not the right type of board}
     EDR_FreeBoardHandle(bh);
     MessageDlg('No PC30G* found',mtError,[mbOk],0);
     halt;
   end;

  EDR_SetDMALevels(bh,5,6);  {Setup DMA transfer}
  EDR_SetADInRange(bh,1,2);  {Setup A/D inputs}

  { setup single channel DMA sampling }
  EDR_SetADChanListLen(bh,0);  {Set number of channels to read}
  for i:=0 to (NumChans - 1) do EDR_AddToADChanList(bh,i);
  EDR_SetADBurstLen(bh,EDR_ADBURSTNUMCHANS);
  EDR_SetADTransferMode(bh,EDR_SINGLEDMA);
  EDR_SetADClockmilliHz(bh,ADClockmilliHz); { 10 kHz = 100 us between bursts }

  EDR_SetDAChanListLen(bh,2);  {Set number of channels to write}
  for i:=0 to 2 do EDR_AddToDAChanList(bh,i); {Add Channels 0 - 2
                                               to the Channel List}

  { Setup program counter/timer and install ISR }
  EDR_CTconfigure(bh,2,2,EDR_CT_BINARY);
  EDR_CTGateSource(bh,2,EDR_CT_ENABLED);
  EDR_CTClockSource(bh,2,EDR_CS_2MHZ);
  EDR_CTWrite(bh,2,TimerCount); { 5kHz = 200 us between ints }

  {Setup IO ports B, C}
  EDR_DIOConfigurePort(bh, PortB, 0, Output);
  EDR_DIOPortOutput(bh, PortB, $20);
  EDR_DIOConfigurePort(bh, PortC, 0, Input);

  {System Startup Parameters}
  CurrentAngle := 0;
```

```
    DeltaAngle := 0;
    DeltaSpeed := 0;
    Torque := 0;
  end;

procedure TSystemController.FormDestroy(Sender: TObject);
begin
  EDR_FreeBoardHandle(bh);
end;

procedure TSystemController.SetUpButtonClick(Sender: TObject);
begin
  SetupForm.Visible := True;
end;

{Start system operation}
procedure TSystemController.StartClick(Sender: TObject);
begin
  SystemStartTimer.Enabled := True;
  SystemStopTimer.Enabled := False;

  {Reset Error Signals}
  Bus4kVError.Checked := False;
  ParBusError.Checked := False;
  AC_ACBusError.Checked := False;
  ParCurrentError.Checked := False;
  SystemCurrentError.Checked := False;

  TorqueSign.ItemIndex := 0;
end;

{Stop system operation}
procedure TSystemController.StopClick(Sender: TObject);
begin
  SystemStartTimer.Enabled := False;
  SystemStopTimer.Enabled := True;
end;

{Shutdown operation - starting mode}
procedure TSystemController.SystemStartTimerTimer(Sender: TObject);
begin
  TorqueStatus.Checked := True;
  SpeedStatus.Checked := True;
  DCDumpStatus.Checked := False;
  SoftStarterStatus.Checked := True;
  DCDriveStatus.Checked := True;
  IF Bus4kVOn.Checked THEN
   begin
    ACMotorDriveStatus.Checked := True;
```

```
      ParInverterStatus.Checked := True;
      IF ParBusOn.Checked AND AC_ACBusOn.Checked THEN
       begin
        ParChopperStatus.Checked := True;
        SystemStartTimer.Enabled := False;
      end;
    end;
  end;

  {Shutdown operation - running mode}
  procedure TSystemController.SystemRunTimerTimer(Sender: TObject);
  begin
   IF Bus4kVError.Checked THEN
     DCDumpTimer.Enabled := True;
   IF ParBusError.Checked OR ParCurrentError.Checked THEN
     ParFault.Enabled := True;
   IF AC_ACBusError.Checked THEN
     AC_ACFault.Enabled := True;
   IF SystemCurrentError.Checked OR DCMotorDriveError.Checked THEN
    begin
     SystemStopTimer.Enabled  := True;
     SystemStartTimer.Enabled := False;
    end;
   IF DCMotorDriveError.Checked THEN
    begin
     DCDriveStatus.Checked := False;
    end;
   IF ParChopperError.Checked OR
     ParInverterError.Checked OR
     ACMotorDriveError.Checked THEN
   begin
    ParInverterStatus.Checked := False;
    ParChopperStatus.Checked := False;
    ACMotorDriveStatus.Checked := False;

    SystemStopTimer.Enabled := True;
    SystemStartTimer.Enabled := False;
  end;
end;

{Shutdown operation - stopping mode}
procedure TSystemController.SystemStopTimerTimer(Sender: TObject);
begin

 SoftStarterStatus.Checked := False;
 DCDumpStatus.Checked := True;
 TorqueStatus.Checked := False;
 SpeedStatus.Checked := False;
 IF NOT Running.Checked THEN
```

```
   begin
    IF NOT Bus4kVOn.Checked THEN
     begin
      DCDriveStatus.Checked := False;
     end;
    IF NOT AC_ACBusOn.Checked THEN
      ACMotorDriveStatus.Checked := False;
   end;

  IF NOT ParBusOn.Checked THEN
   begin
    ParInverterStatus.Checked := False;
    ParChopperStatus.Checked := False;
   end;

end;

{Shutdown operation - par fault detected mode}
procedure TSystemController.ParFaultTimer(Sender: TObject);
begin
  ParInverterStatus.Checked := False;
  ParChopperStatus.Checked := False;
  IF NOT ParBusError.Checked THEN
    ParFault.Enabled := False;
end;

{Shutdown operation - synchrnous drive fault detected mode}
procedure TSystemController.AC_ACFaultTimer(Sender: TObject);
begin
  ACMotorDriveStatus.Checked := False;
  IF NOT AC_ACBusError.Checked THEN
    AC_ACFault.Enabled := False;
end;

{Shutdown operation - dc dump mode}
procedure TSystemController.DcDumpTimerTimer(Sender: TObject);
begin
  DCDumpStatus.Checked := True;
  IF NOT Bus4kVError.Checked THEN
   begin
    DCDumpStatus.Checked := False;
    DCDumpTimer.Enabled := False;
   end;
end;

{Shutdown operation - write system stati}
procedure TSystemController.StatusWriterTimer(Sender: TObject);
var
  DCDumpS : Integer;
```

```
   ParChopS : Integer;
   ParInvS : Integer;
   ACDriveS : Integer;
   DCDriveS : Integer;
 begin
   IF DCDumpStatus.Checked THEN DCDumpS := 1 ELSE DCDumpS := 0;
   IF ParChopperStatus.Checked THEN ParChopS := 1 ELSE ParChopS := 0;
   IF ParInverterStatus.Checked THEN ParInvS := 1 ELSE ParInvS := 0;
   IF ACMotorDriveStatus.Checked THEN ACDriveS := 1 ELSE ACDriveS := 0;
   IF DCDriveStatus.Checked THEN DCDriveS := 1 ELSE DCDriveS := 0;
   EDR_DIOLineOutput(bh, PortB, 0, ParInvS);
   EDR_DIOLineOutput(bh, PortB, 1, ACDriveS);
   EDR_DIOLineOutput(bh, PortB, 2, DCDriveS);
   EDR_DIOLineOutput(bh, PortB, 3, ParChopS);
   EDR_DIOLineOutput(bh, PortB, 7, DCDumpS);
end;

{Shutdown operation - read system faults}
procedure TSystemController.FaultReaderTimer(Sender: TObject);
var
  ParChopF : Integer;
  ParInvF : Integer;
  ACDriveF : Integer;
  DCDriveF : Integer;
begin
  EDR_DIOLineInput(bh, PortC, 0, ParInvF);
  EDR_DIOLineInput(bh, PortC, 1, ACDriveF);
  EDR_DIOLineInput(bh, PortC, 2, DCDriveF);
  EDR_DIOLineInput(bh, PortC, 3, ParChopF);

  IF DCDriveF = 0 THEN begin
            DCMotorDriveError.Checked := True;
            DCMotorDriveError.Color := clRed;
            end
            ELSE begin
            DCMotorDriveError.Checked := False;
            DCMotorDriveError.ParentColor := True
            end;
  IF ACDriveF = 0 THEN begin
            ACMotorDriveError.Checked := True;
            ACMotorDriveError.Color := clRed;
            end
            ELSE begin
            ACMotorDriveError.Checked := False;
            ACMotorDriveError.ParentColor := True
            end;
  IF ParChopF = 0 THEN begin
            ParChopperError.Checked := True;
            ParChopperError.Color := clRed;
```

```
                end
                ELSE begin
                ParChopperError.Checked := False;
                ParChopperError.ParentColor := True
                end;
      IF ParInvF = 0 THEN begin
                ParInverterError.Checked := True;
                ParInverterError.Color := clRed;
                end
                ELSE begin
                ParInverterError.Checked := False;
                ParInverterError.ParentColor := True
                end;
end;

{Shutdown operation - Write references to drives}
procedure TSystemController.DataWriterTimer(Sender: TObject);
var
  ScaledSpeed : Integer;
  TorqueOut : Integer;
  Angle_uVolts : Integer;
  DSpeed_uVolts : Integer;
  Torque_uVolts : Integer;

begin
  {Write Firing Angle to the Soft-Starter}
  Angle_uVolts := 10000000 - Trunc((Angle * 10000000) / 180);
  EDR_DAOutVoltage(bh, 0, Angle_uVolts);

   {Write Out Delta Speed}
  DSpeed_uVolts := Trunc((DeltaSpeed + 500) * 10000);
  IF DSpeed_uVolts > 10000000 then DSpeed_uVolts := 10000000
  else IF DSpeed_uVolts < 0 then DSpeed_uVolts := 0;
  EDR_DAOutVoltage(bh, 1, DSpeed_uVolts);
  {label50.caption := IntToStr(DSpeed_uVolts);}

   {Write Out Torque}
  Torque_uVolts := Trunc(Torque * 39215);
  EDR_DAOutVoltage(bh, 2, Torque_uVolts);

  EDR_DIOConfigurePort(bh, PortA, 0, Output);

  {Write Speed Reference to the DC Motor Drive}
  EDR_DIOLineOutput(bh, PortB, 5, 1);
  IF SpeedStatus.Checked = True then SpeedOut := CurrentSpeed + Trunc(DeltaSpeed)
                          else SpeedOut := 0;
  ScaledSpeed := Trunc(SpeedOut*255 / MaxRPM);
  EDR_DIOPortOutput(bh, PortA, ScaledSpeed);
  EDR_DIOLineOutput(bh, PortB, 5, 0);
```

```
  {Write Torque Reference to the AC Motor Drive}
  EDR_DIOLineOutput(bh, PortB, 4, 1);
  IF NOT TorqueStatus.Checked then Torque := 127;
  EDR_DIOPortOutput(bh, PortA, Trunc(Torque));
  EDR_DIOLineOutput(bh, PortB, 4, 0);
end;

{Shutdown operation - read system parameters}
procedure TSystemController.DataReaderTimer(Sender: TObject);
var SpeedIn : Integer;
begin
  {Read System Parameters}
  EDR_ADInOneVoltage(bh, 0, Ia);
  EDR_ADInOneVoltage(bh, 1, Ib);
  EDR_ADInOneVoltage(bh, 2, Ic);
  EDR_ADInOneVoltage(bh, 3, V900);
  EDR_ADInOneVoltage(bh, 4, V4kV);
  EDR_ADInOneVoltage(bh, 5, V800);
  EDR_ADInOneVoltage(bh, 6, Ipar);

  {Evaluate System Parameters}
  {System Currents}
  IF (Ia > CurrentOnLimit) OR
    (Ib > CurrentOnLimit) OR
    (Ic > CurrentOnLimit) then SystemCurrentOn.Checked := True
                 else SystemCurrentOn.Checked := False;
  IF (Ia > CurrentErrorLimit) OR
    (Ib > CurrentErrorLimit) OR
    (Ic > CurrentErrorLimit) then SystemCurrentError.Checked := True;

  {900V Bus Voltage}
  IF V900 > V900BusOnLimit then AC_ACBusOn.Checked := True
                 else AC_ACBusOn.Checked := False;
  IF V900 > V900BusErrorLimit then AC_ACBusError.Checked := True;

  {4 kV Bus Voltage}
  IF V4kV > V4kVBusOnLimit then Bus4kVOn.Checked := True
                 else Bus4kVOn.Checked := False;
  IF V4kV > V4kVBusErrorLimit then
   begin
    Bus4kVError.Checked := True;
    DCDumpTimer.Enabled := True;
   end;

  {800V Bus Voltage}
  IF V800 > V800BusOnLimit then ParBusOn.Checked := True
                 else ParBusOn.Checked := False;
  IF V800 > V800BusErrorLimit then ParBusError.Checked := True;
```

```
    {Parallel Current}
    IF Ipar > ParallelCurrentOnLimit then ParCurrentOn.Checked := True
                        else ParCurrentOn.Checked := False;
    IF Ipar > ParallelCurrentErrorLimit then ParCurrentError.Checked := True;

    {Read Current Motor Speed}
    EDR_DIOConfigurePort(bh, PortA, 0, Input);
    EDR_DIOLineOutput(bh, PortB, 6, 0);
    EDR_DIOPortInput(bh, PortA, SpeedIn);
    CurrentSpeed := SpeedIn*MaxRpm / 255;
    EDR_DIOLineOutput(bh, PortB, 6, 1);

    {Evaluate Speed(RPM) for Protection Purposes}
    IF CurrentSpeed > RunningSpeed then Running.Checked := True
                        else Running.Checked := False;
   Speed.Progress  := (SpeedIn * 100) div 255;
  end;

  {Fuzzy operation}
  procedure TSystemController.FuzzificationTimer(Sender: TObject);
  var
   IaHigh, IbHigh, IcHigh : Real;
   IaVeryHigh, IbVeryHigh, IcVeryHigh : Real;
   Bus4kVLow, Bus4kVMedium, Bus4kVRight, Bus4kVHigh : Real;
   Bus900VLow, Bus900VRight, Bus900VHigh : Real;
   Bus800VRight, Bus800VHigh : Real;
   ParallelCurrentHigh : Real;
   SpeedHigh : Real;
  begin
   {Fuzzification of Input Variables}
   IaHigh := HighCurrent(Ia);
  { Gauge1.Progress := Trunc(IaHigh); }
   IbHigh := HighCurrent(Ib);
  { Gauge2.Progress := Trunc(IbHigh); }
   IcHigh := HighCurrent(Ic);
  { Gauge3.Progress := Trunc(IcHigh);}

   IaVeryHigh := VeryHighCurrent(Ia);
  { Gauge14.Progress := Trunc(IaVeryHigh); }
   IbVeryHigh := VeryHighCurrent(Ib);
  { Gauge19.Progress := Trunc(IbVeryHigh); }
   IcVeryHigh := VeryHighCurrent(Ic);
  { Gauge20.Progress := Trunc(IcVeryHigh); }

   Bus900VLow   := LowSynchroBus(V900);
  {Gauge4.Progress := Trunc(Bus900VLow);   }
   Bus900VRight := RightSynchroBus(V900);
  { Gauge5.Progress := Trunc(Bus900VRight);   }
```

```
Bus900VHigh  := HighSynchroBus(V900);
{ Gauge6.Progress := Trunc(Bus900VHigh);  }

Bus800VRight := RightParallelBus(V800);
{  Gauge7.Progress := Trunc(Bus800VRight); }
Bus800VHigh  := HighParallelBus(V800);
{  Gauge8.Progress := Trunc(Bus800VHigh);  }

ParallelCurrentHigh := HighParallelCurrent(Ipar);
{ Gauge9.Progress := Trunc(ParallelCurrentHigh); }

Bus4kVLow    := Low4kVbus(V4kV);
{ Gauge10.Progress := Trunc(Bus4kVLow); }
Bus4kVMedium := Medium4kVBus(V4kV);
{ Gauge11.Progress := Trunc(Bus4kVMedium); }
Bus4kVRight  := Right4kVBus(V4kV);
{ Gauge12.Progress := Trunc(Bus4kVRight);  }
Bus4kVHigh   := High4kVBus(V4kV);
{ Gauge13.Progress := Trunc(Bus4kVHigh);   }

SpeedHigh := HighSpeed(Trunc(CurrentSpeed));

{System Control Inputs}
DeltaScreen  := StrToInt(SpeedRequired.Text) - CurrentSpeed;
ScreenTorque := ((1-2*TorqueSign.ItemIndex)*TorqueProgressSize + 100)*255
            div 200;

{Determine Alphas(Degrees of Truth) for fuzzy rules}

IF IaHigh > IbHigh then AlphaACCurrentsHigh := IaHigh
             else AlphaACCurrentsHigh := IbHigh;
IF Ichigh > AlphaACCurrentsHigh then AlphaACCurrentsHigh := IcHigh;

{Rule1 : IF low(900V) OR High(V4kV) OR High(800V)
         OR High(Ia) THEN ...}
IF Bus900VLow > Bus4kVHigh  then Alpha1 := Bus900VLow
                   else Alpha1 := Bus4kVHigh;
IF Alpha1 < Bus800VHigh then Alpha1 := Bus800VHigh;
IF Alpha1 < AlphaACCurrentsHigh then Alpha1 := AlphaACCurrentsHigh;

{Rule2 : IF Med(V4kV) OR High(900V) OR Low(800V)
         OR (High(Ia) OR High(Ib) OR High(Ic)) THEN ...}
IF Bus4kVMedium > Bus900VHigh then Alpha2 := Bus4kVMedium
                    else Alpha2 := Bus900VHigh;
IF Alpha2 < AlphaACCurrentsHigh then Alpha2 := AlphaACCurrentsHigh;

{Rule3 : IF High(Speed) THEN ...}
Alpha3 := SpeedHigh;
```

```
{Rule4 : IF NOT (VeryHigh(Ia) OR VeryHigh(Ib) OR VeryHigh(Ic)) THEN ...}
Alpha4 := 100 - Alpha5;

{Rule5 : IF VeryHigh(Ia) OR VeryHigh(Ib) OR VeryHigh(Ic) THEN ...}
IF IaVeryHigh > IbVeryHigh then Alpha5 := IaVeryHigh
                 else Alpha5 := IbVeryHigh;
IF Alpha5 < IcVeryHigh then Alpha5 := IcVeryHigh;

{Rule6 : IF Right(800V) AND Right(900V) AND NOT Low(4kV) AND NOT Medium(4kV)
      AND NOT High(IPar) AND NOT High(Ia,Ib,Ic)}
IF Bus800VRight < Bus900VRight then Alpha6 := Bus800VRight
                   else Alpha6 := Bus900VRight;
IF Alpha6 > (100 - Bus4kVLow) then Alpha6 := (100 - Bus4kVLow);
IF Alpha6 > (100 - Bus4kVMedium) then Alpha6 := (100 - Bus4kVMedium);
IF Alpha6 > (100 - ParallelCurrentHigh)
       then Alpha6 := (100 - ParallelCurrentHigh);
IF Alpha6 > (100 - AlphaACCurrentsHigh)
       then Alpha6 := (100 - AlphaACCurrentsHigh);

{Open for further expansion of the rule base}


{Use Centre Of Maximum (COM) defuzzification}
{Angle}
IF (Alpha4 + Alpha5) > 0 then
  Angle := (Alpha5 * AngleZero + Alpha4 * CurrentAngle) /
        (Alpha4 + Alpha5);

{DeltaAngle}
IF (Alpha4 + Alpha5) > 0 then
  DeltaAngle := ( Alpha5*DeltaAngleMedNeg +
           Alpha4*DeltaAngleMedPos) /
          (Alpha4 + Alpha5)
else
  DeltaAngle := DeltaAngleMedNeg;

IF SoftStarterStatus.Checked then CurrentAngle := CurrentAngle + DeltaAngle
                 else CurrentAngle := 0;
IF CurrentAngle > 180 then CurrentAngle := 180;
IF CurrentAngle < 0 then CurrentAngle := 0;

{DeltaSpeed}
IF (Alpha1 + Alpha2 + Alpha3 + Alpha6) > 0 then
  DeltaSpeed := ((Alpha1 + Alpha2)*DeltaSpeedLow +
          Alpha3*DeltaSpeedMedNeg +
          Alpha6*DeltaScreen) /
         (Alpha1 + Alpha2 + Alpha3 + Alpha6)
else
  DeltaSpeed := 0;
```

```
  {Torque}
  IF (Alpha1 + Alpha2 + Alpha6) > 0 then
    Torque    := ((Alpha1 + Alpha2)*TorqueLow +
                 Alpha6*ScreenTorque) /
                (Alpha1 + Alpha2 + Alpha6)
  else
    Torque := TorqueLow;

  {Display Fuzzy Outputs}
  Gauge15.Progress := Trunc(Angle);
  Gauge16.Progress := Trunc(DeltaAngle);
  Gauge17.Progress := Trunc(DeltaSpeed);
  Gauge18.Progress := Trunc(Torque);
end;

procedure TSystemController.HelperTimer(Sender: TObject);
begin
  {Set Speed Gauge}
  IF StrToInt(SpeedRequired.Text) > MaxRPM THEN
    SpeedRequired.Text := IntToStr(MaxRPM);
  SpeedRefGauge.Progress := StrToInt(SpeedRequired.Text)* 100 div MaxRPM;

  {Set Torque Gauge}
  IF StrToInt(TorqueRequired.Text) > 100 THEN
    TorqueRequired.Text :=  IntToStr(100);

  TorqueProgressSize := StrToInt(TorqueRequired.Text);
  TorqueRefGauge.Progress := (1-2*TorqueSign.ItemIndex)*TorqueProgressSize;

end;

end.

{Connection table

AD Chan 0  - Ia
AD Chan 1  - Ib
AD Chan 2  - Ic
AD Chan 3  - 900V bus
AD Chan 4  - 4 kV bus
AD Chan 5  - 800V bus
AD Chan 6  - I parallel

DA Chan 0  - Angle

Out[0..7]  - Port A[0..7]

Parallel Inverter Status - Port B 0
```

Simulator Status - Port B 1
Motor Drive Status - Port B 2
Parallel Chopper Status - Port B 3

Torque Write - Port B 4
Speed Write - Port B 5
Speed Read - Port B 6
DC Dump Status - Port B 7

Parallel Inverter Fault - Port C 0
Simulator Fault - Port C 1
Motor Drive Status - Port C 2
Parallel Chopper Fault - Port C 3

}

Appendix D2 - Delphi system control program - Board set-up window

```
unit Setup;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, StdCtrls, EDR32, Fuzzy;

type
  TSetupForm = class(TForm)
    MainMenu1: TMainMenu;
    Front1: TMenuItem;
    Channel11: TMenuItem;
    Channel21: TMenuItem;
    Channel31: TMenuItem;
    Channel41: TMenuItem;
    Channel42: TMenuItem;
    Channel61: TMenuItem;
    Channel71: TMenuItem;
    Channel81: TMenuItem;
    Channel91: TMenuItem;
    Channel92: TMenuItem;
    Channel101: TMenuItem;
    Channel111: TMenuItem;
    Channel121: TMenuItem;
    Channel131: TMenuItem;
    Channel141: TMenuItem;
    Channel151: TMenuItem;
    OutPutChannelParamaters1: TMenuItem;
    Channel12: TMenuItem;
    Channel22: TMenuItem;
    Channel32: TMenuItem;
    Channel43: TMenuItem;
    INChanParams: TGroupBox;
    OutChanParams: TGroupBox;
    ADInType: TComboBox;
    ADInRange: TComboBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel,
    Label4: TLabel;
```

```
    DAOutRange: TComboBox;
    Label5: TLabel;
    ReadADParams: TButton;
    Label7: TLabel;
    ChannelName: TLabel;
    OutChannelName: TLabel;
    ReadDAParams: TButton;
    InGain: TComboBox;
    OutGain: TEdit;
    procedure Channel11Click(Sender: TObject);
    procedure Channel21Click(Sender: TObject);
    procedure Channel71Click(Sender: TObject);
    procedure Channel31Click(Sender: TObject);
    procedure Channel41Click(Sender: TObject);
    procedure Channel42Click(Sender: TObject);
    procedure Channel61Click(Sender: TObject);
    procedure Channel81Click(Sender: TObject);
    procedure Channel91Click(Sender: TObject);
    procedure Channel92Click(Sender: TObject);
    procedure Channel101Click(Sender: TObject);
    procedure Channel111Click(Sender: TObject);
    procedure Channel121Click(Sender: TObject);
    procedure Channel131Click(Sender: TObject);
    procedure Channel141Click(Sender: TObject);
    procedure Channel151Click(Sender: TObject);
    procedure INChanParamsDblClick(Sender: TObject);
    procedure OutChanParamsDblClick(Sender: TObject);
    procedure InGainChange(Sender: TObject);
    procedure ReadADParamsClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ADInRangeChange(Sender: TObject);
    procedure ADInTypeChange(Sender: TObject);
    procedure Channel12Click(Sender: TObject);
    procedure Channel22Click(Sender: TObject);
    procedure Channel32Click(Sender: TObject);
    procedure Channel43Click(Sender: TObject);
    procedure OutGainChange(Sender: TObject);
    procedure ReadDAParamsClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  SetupForm: TSetupForm;

  bh        : Integer;
  CurrentChan : Integer;
```

```
  CurrentGain : Integer;
  CurrentRange : Integer;

implementation

{$R *.DFM}

procedure TSetupForm.Channel11Click(Sender: TObject);
begin
 CurrentChan := 0;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;
end;

procedure TSetupForm.Channel21Click(Sender: TObject);
begin
 CurrentChan := 1;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;
end;

procedure TSetupForm.Channel31Click(Sender: TObject);
begin
 CurrentChan := 2;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel41Click(Sender: TObject);
begin
 CurrentChan := 3;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel42Click(Sender: TObject);
begin
 CurrentChan := 4;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel61Click(Sender: TObject);
begin
 CurrentChan := 5;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
```

```
  InChanParams.Visible := True;

end;

procedure TSetupForm.Channel71Click(Sender: TObject);
begin
 CurrentChan := 6;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;
end;

procedure TSetupForm.Channel81Click(Sender: TObject);
begin
 CurrentChan := 7;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel91Click(Sender: TObject);
begin
 CurrentChan := 8;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel92Click(Sender: TObject);
begin
 CurrentChan := 9;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel101Click(Sender: TObject);
begin
 CurrentChan := 10;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;

end;

procedure TSetupForm.Channel111Click(Sender: TObject);
begin
 CurrentChan := 11;
 ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
 InChanParams.Visible := True;
```

```
end;

procedure TSetupForm.Channel121Click(Sender: TObject);
begin
  CurrentChan := 12;
  ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  InChanParams.Visible := True;

end;

procedure TSetupForm.Channel131Click(Sender: TObject);
begin
  CurrentChan := 13;
  ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  InChanParams.Visible := True;

end;

procedure TSetupForm.Channel141Click(Sender: TObject);
begin
  CurrentChan := 14;
  ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  InChanParams.Visible := True;

end;

procedure TSetupForm.Channel151Click(Sender: TObject);
begin
  CurrentChan := 15;
  ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  InChanParams.Visible := True;

end;

procedure TSetupForm.INChanParamsDblClick(Sender: TObject);
begin
  INChanParams.Visible := False;
end;

procedure TSetupForm.OutChanParamsDblClick(Sender: TObject);
begin
  OutChanParams.Visible := False;
end;

procedure TSetupForm.InGainChange(Sender: TObject);
var
  Gain : integer;
  Code : Integer;
begin
```

```
    Gain := StrToInt(InGain.Text);
    Code := EDR_SetADInGain(bh, CurrentChan, Gain);
    EDR_GetADInGain(bh, CurrentChan, Gain);
    ChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
    Label7.Caption := 'Label7 = ' + IntToStr(Code);
  end;

  procedure TSetupForm.ReadADParamsClick(Sender: TObject);
  var
    Gain : Integer;
    Range : Integer;
    ADType : Integer;
  begin
   EDR_GetADInGain(bh, CurrentChan, Gain);
   EDR_GetADInRange(bh, CurrentChan, Range);
   EDR_GetADInType(bh, CurrentChan, ADType);

   InGain.Text := IntToStr(Gain);
   ADInRange.ItemIndex := Range;
   ADInType.ItemIndex := ADType;
  end;

  procedure TSetupForm.FormCreate(Sender: TObject);
  begin
   bh := SystemController.Tag;
   InChanParams.Visible := False;
   OutChanParams.Visible := False;
  end;

  procedure TSetupForm.ADInRangeChange(Sender: TObject);
  begin
   EDR_SetADInRange(bh, CurrentChan, ADInRange.ItemIndex);
  end;

  procedure TSetupForm.ADInTypeChange(Sender: TObject);
  begin
   EDR_SetADInType(bh, CurrentChan, ADInType.ItemIndex);
  end;

  procedure TSetupForm.Channel12Click(Sender: TObject);
  begin
   CurrentChan := 0;
   OutChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
   OutChanParams.Visible := True;
  end;

  procedure TSetupForm.Channel22Click(Sender: TObject);
  begin
   CurrentChan := 1;
```

```
  OutChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  OutChanParams.Visible := True;
end;

procedure TSetupForm.Channel32Click(Sender: TObject);
begin
  CurrentChan := 2;
  OutChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  OutChanParams.Visible := True;
end;

procedure TSetupForm.Channel43Click(Sender: TObject);
begin
  CurrentChan := 3;
  OutChannelName.Caption := 'Channel ' + IntToStr(CurrentChan);
  OutChanParams.Visible := True;
end;

procedure TSetupForm.OutGainChange(Sender: TObject);
var Code : integer;
begin
  Code := EDR_SetDAOutGain(bh, CurrentChan, StrToInt(OutGain.Text));
  Label7.Caption := 'Label7 = ' + IntToStr(Code);
end;

procedure TSetupForm.ReadDAParamsClick(Sender: TObject);
var
   Gain : Integer;
   Range : Integer;
begin
  EDR_GetDAOutGain(bh, CurrentChan, Gain);
  EDR_GetDAOutRange(bh, CurrentChan, Range);

  OutGain.Text := IntToStr(Gain);
  DAOutRange.ItemIndex := Range;
end;

end.
```

Appendix D3 - Delphi system control program - Program Implementation

```
program FuzzyPr;

uses
  Forms,
  Setup in '\USERS\EBeaud\DELPHI\FuzzyS\Setup.pas' {SetupForm},
  Fuzzy in '\USERS\EBeaud\DELPHI\FuzzyS\Fuzzy.pas' {SystemController};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TSystemController, SystemController);
  Application.CreateForm(TSetupForm, SetupForm);
  Application.Run;
end.
```

Appendix E

Rundown Test Characterising the MG set

Appendix E - Rundown Test characterising the MG set

The purpose of the rundown test is to determine the inertia of the rotor of the MG set. When the motor is allowed to run down freely the only force stopping the rotor is the friction. The equation governing the rundown of the rotor is

$$B \cdot \omega = -J \frac{d\varpi}{dt} \qquad \text{.......1}$$

with the speed in radians per second. The speed was measured during rundown vs. time so the only parameter that is missing is the friction of the rotor. A method of determining the friction at a certain operating speed is measuring the power consumption of the motor while constantly running at that speed. The power is then dissipated in the resistance of the conductors and in the friction of the rotor. The resistance was measured by disconnecting the field and applying a voltage on the armature winding. The measured resistance was 4.17 Ω.

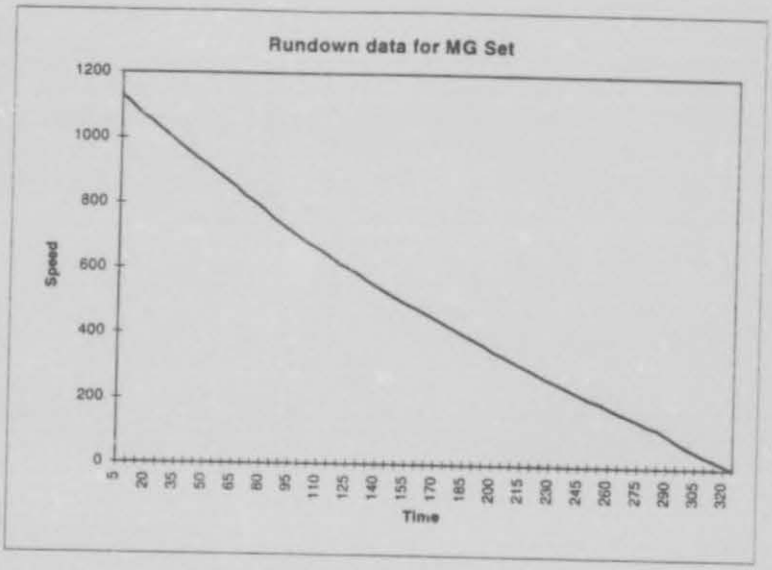
The motor was run up to 1000 rpm with a dc voltage of 1300 V drawing 0.9 amp. The power loss in the conductors is then 3.38 W. The rest of the 1170 W is dissipated in the rotor through friction. The power dissipated is related to the friction according to equation 2.

$$P_{Loss} = B \cdot \varpi^2 \qquad \text{.......2}$$

From equation 2 the friction constant B was calculated as 0.11 Nms/rad. The data for the rundown test is given at the end of this appendix. To calculate the inertia the change in speed has to be found around an operating point where the friction is known. The rate of change around the 1000 rpm operating point was calculated to be 0.5 rad/s^{-2}. The inertia is then calculated as 23 kg.m^2.

The energy stored in the rotor is $E_{Stored} = \frac{1}{2} J\varpi^2$. At a maximum speed of 1500 rpm (157 rad/s) the energy stored is 284 kJ.

0	1156
5	1129
10	1103
15	1072
20	1053
25	1028
30	1004
35	980
40	958
45	935
50	917
55	894
60	872
65	850
70	823
75	805
80	782
85	755
90	734
95	713
100	692
105	673
110	656
115	637
120	614
125	600
130	582
135	560
140	543
145	526
150	509
155	492
160	479
165	463
170	447
175	431
180	415
185	399
190	384
195	369
200	350
205	336
210	321
215	306
220	293
225	276
230	262
235	249
240	235
245	221
250	207
255	197
260	181
265	168
270	156
275	143
280	129
285	120
290	102
295	83
300	67
305	52
310	37
315	25
320	12



Appendix F

Altera EPLD system Control Program

Appendix F - Altera EPLD system control program

```
% ----------------------------------------- %
% -- Protection Control                 -- %
% -- E.Beaud                            -- %
% -- 04/11/1996                         -- %
% -- Ver 1.2                            -- %
% ----------------------------------------- %

SUBDESIGN decoder6
(
        clk             : INPUT;
        AC_ACF          : INPUT;
        ParallelF       : INPUT;
        DCDriveF        : INPUT;

        900VF           : INPUT;
        800VF           : INPUT;
        4kVF            : INPUT;
        900VOn          : INPUT;
        800VOn          : INPUT;
        4kVOn           : INPUT;

        IParChop        : INPUT;
        SysReset        : INPUT;
        EOC             : INPUT;
        Bus[7..0]       : INPUT;
        Ia_Ib_Ic        : INPUT;

        ParInvertS      : OUTPUT;
        ParChopS        : OUTPUT;
        DCDriveS        : OUTPUT;
        AC_ACS          : OUTPUT;
        SoftStartS      : OUTPUT;
        DcDumpS         : OUTPUT;

        ALE             : OUTPUT;
        A/D_OE          : OUTPUT;
        STC             : OUTPUT;
        SpeedR          : OUTPUT;
        Write           : OUTPUT;
        Speed/TorqueS   : OUTPUT;
        Speed/Torque    : OUTPUT;
        Clock2          : OUTPUT;
```

```
)

VARIABLE
        ParInvertS              : DFFE;
        ParChopS                : DFFE;
        DCDriveS                : DFFE;
        AC_ACS                  : DFFE;
        SoftStartS              : DFFE;
        DcDumpS                 : DFFE;
        SpeedS                  : DFFE;
        TorqueS                 : DFFE;
        OldMonitor[2..1]        : DFFE;
        SpeedB[7..0]            : DFFE;
        Running                 : DFFE;
        Speed/Torque            : DFFE;
        Speed/TorqueS           : DFFE;
        Fault                   : DFFE;
        A/D_OE, STC             : DFFE;
        ALE                     : DFFE;
        Clock2                  : DFFE;
        ClockCounter            : DFFE;
        Reset                   : DFFE;

        Monitor         : MACHINE
                                OF Bits (q[5..1])
                                WITH STATES
        (
                        AllOff          = B"00000",
                        AllOk           = B"00001",
                        WaitS0          = B"00010",
                        WaitBuses       = B"00011",
                        BeginW1         = B"00100",
                        BeginW2         = B"00101",
                        EndW1           = B"00110",
                        EndW2           = B"00111",
                        EndW3           = B"01000",
                        EndW4           = B"01001",
                        DCDrive         = B"01010",
                        Parallel        = B"01011",
                        Current         = B"01100",
                        Bus900          = B"01101",
                        Bus800          = B"01110",
                        SwitchOn        = B"01111",
                        SpeedR1         = B"10000",
                        SpeedR2         = B"10001",
                        SpeedR3         = B"10010",
                        Illegal1        = B"10011",
                        Illegal2        = B"10100",
                        Illegal3        = B"10101",
```

```
                    Illegal4        = B"10110",
                    Illegal5        = B"10111",
                    Illegal6        = B"11000",
                    Illegal7        = B"11001",
                    Illegal8        = B"11010",
                    Illegal9        = B"11011",
                    Illegal10       = B"11100",
                    Illegal11       = B"11101",
                    Illegal12       = B"11110",
                    Illegal13       = B"11111"
        );

BEGIN
    DEFAULTS
        SpeedR                  = vcc;
        Write                   = gnd;
        ParInvertS.ena          = gnd;
        ParChopS.ena            = gnd;
        DCDriveS.ena            = gnd;
        AC_ACS.ena              = gnd;
        SoftStartS.ena          = gnd;
        DcDumpS.ena             = gnd;
        SpeedS.ena              = gnd;
        TorqueS.ena             = gnd;
        OldMonitor[].ena        = gnd;
        SpeedB[7..0].ena        = gnd;
        Running.ena             = gnd;
        Speed/Torque.ena        = gnd;
        Speed/TorqueS.ena       = gnd;
        Fault.ena               = gnd;
        A/D_OE.ena              = gnd;
        ALE.ena                 = gnd;
        STC.ena                 = gnd;
        Clock2.ena              = gnd;
        ClockCounter.ena        = gnd;
        Reset.ena               = gnd;
    END DEFAULTS;

    Monitor.clk         = clk;
    ParInvertS.clk      = clk;
    ParChopS.clk        = clk;
    DCDriveS.clk        = clk;
    AC_ACS.clk          = clk;
    SoftStartS.clk      = clk;
    DcDumpS.clk         = clk;
    SpeedS.clk          = clk;
    TorqueS.clk         = clk;
    OldMonitor[].clk    = clk;
    SpeedB[7..0].clk    = clk;
```

```
        Running.clk             = clk;
        Speed/Torque.clk        = clk;
        Speed/TorqueS.clk       = clk;
        Fault.clk               = clk;
        A/D_OE.clk              = clk;
        ALE.clk                 = clk;
        STC.clk                 = clk;
        Clock2.clk              = clk;
        ClockCounter.clk        = clk;
        Reset.clk               = clk;

        IF SysReset THEN
                Reset.ena               = vcc;
                Reset.d                 = vcc;
        ELSE
                Reset.ena               = vcc;
                Reset.d                 = gnd;
        END IF;

        IF CLockCounter.q THEN
                Clock2.ena              = vcc;
                Clock2.d                = !Clock2.q;

                CLockCounter.ena        = vcc;
                CLockCounter.d          = gnd;
        ELSE
                CLockCounter.ena        = vcc;
                CLockCounter.d          = vcc;
        END IF;

CASE Monitor IS
        WHEN AllOk =>
                OldMonitor[].ena        = vcc;
                OldMonitor[].d          = Monitor.q[2..1];

                IF Reset.q THEN
                        Monitor = AllOff;
                ELSIF !ParallelF OR !AC_ACF OR !ParChop THEN
                        Monitor         = Parallel;
                ELSIF !DCDriveF THEN
                        Monitor         = DCDrive;
                ELSIF Ia_Ib_Ic THEN
                        Monitor         = Current;
                ELSIF 900VF THEN
                        Monitor         = Bus900;
                ELSIF 800VF THEN
                        Monitor         = Bus800;
                ELSIF 4KVF THEN
                        DCDumpS.ena             = vcc;
```

```
            DCDumpS.d             = vcc;
        ELSIF !EOC AND STC.q THEN
            STC.ena               = vcc;
            ALE.ena               = vcc;
            Speed/Torque.ena      = vcc;
            STC.d                 = gnd;
            ALE.d                 = gnd;
            Speed/Torque.d        = !Speed/Torque.q;
        ELSIF EOC THEN
            Monitor        = EndW1;
        ELSE
            IF !Fault.q THEN
                DCDumpS.ena       = vcc;
                DCDumpS.d         = gnd;
            END IF;
            Monitor               = SpeedR1;
        END IF;

    WHEN ENDW1 =>
        A/D_OE.ena            = vcc;
        A/D_OE.d              = vcc;
        Monitor               = ENDW2;

    WHEN ENDW2 =>
        Monitor               = EndW3;

    WHEN ENDW3 =>
        Write                 = vcc;
        Monitor               = ENDW4;

    WHEN ENDW4 =>
        Write                 = vcc;
        Monitor               = BeginW1;

    WHEN BeginW1 =>
        IF Speed/Torque.q THEN
            Speed/TorqueS.ena     = vcc;
            Speed/TorqueS.d       = !SpeedS.q;
        ELSE
            Speed/TorqueS.ena     = vcc;
            Speed/TorqueS.d       = !TorqueS.q;
        END IF;
        A/D_OE.ena            = vcc;
        ALE.ena               = vcc;

        A/D_OE.d              = gnd;
        ALE.d                 = vcc;
        Monitor               = BeginW2;
```

```
WHEN BeginW2 =>
        STC.ena             = vcc;
        STC.d               = vcc;
        IF !OldMonitor2.q THEN
                Monitor     = AllOk;
        ELSE
                IF OldMonitor1.q THEN
                        Monitor = WaitBuses;
                ELSE
                        Monitor = WaitS0;
                END IF;
        END IF;

WHEN SpeedR1 =>
        SpeedR        = gnd;
        Monitor       = SpeedR2;

WHEN SpeedR2 =>
        SpeedR        = gnd;
        SpeedB[7..0].ena = vcc;
        SpeedB[].d    = Bus[];
        IF SpeedB[] < H"2" THEN
                Running.ena   = vcc;
                Running.d     = gnd;
        ELSE
                Running.ena   = vcc;
                Running.d     = vcc;
        END IF;
        Monitor       = SpeedR3;

WHEN SpeedR3 =>
        IF !OldMonitor2.q THEN
                Monitor = AllOk;
        ELSE
                IF OldMonitor1.q THEN
                        Monitor = WaitBuses;
                ELSE
                        Monitor = WaitS0;
                END IF;
        END IF;

WHEN Parallel =>
        Fault.ena       = vcc;
        Fault.d         = vcc;

        ParInvertS.ena          = vcc;
        ParChopS.ena            = vcc;
        AC_ACS.ena              = vcc;
        SoftStartS.ena          = vcc;
```

```
            DcDumpS.ena         = vcc;
            SpeedS.ena          = vcc;

            ParInvertS.d   = gnd;
            ParChopS.d     = gnd;
            AC_ACS.d       = gnd;
            SoftStartS.d   = gnd;
            DcDumpS.d      = vcc;
            SpeedS.d       = gnd;
            Monitor        = WaitS0;

    WHEN DCDrive =>
            Fault.ena      = vcc;
            Fault.d        = vcc;

            ParInvertS.ena      = vcc;
            ParChopS.ena        = vcc;
            DCDriveS.ena        = vcc;
            SoftStartS.ena      = vcc;
            DcDumpS.ena         = vcc;
            TorqueS.ena         = vcc;

            ParInvertS.d   = gnd;
            ParChopS.d     = gnd;
            DCDriveS.d     = gnd;
            SoftStartS.d   = gnd;
            DcDumpS.d      = vcc;
            TorqueS.d      = gnd;
            Monitor        = WaitS0;

    WHEN Current =>
            Fault.ena      = vcc;
            Fault.d        = vcc;

            ParInvertS.ena      = vcc;
            ParChopS.ena        = vcc;
            SoftStartS.ena      = vcc;
            DcDumpS.ena         = vcc;
            SpeedS.ena          = vcc;
            TorqueS.ena         = vcc;

            ParInvertS.d   = gnd;
            ParChopS.d     = gnd;
            SoftStartS.d   = gnd;
            DcDumpS.d      = vcc;
            SpeedS.d       = gnd;
            TorqueS.d      = gnd;
            Monitor        = WaitS0;
```

```
WHEN Bus800 =>
        Fault.ena       = vcc;
        Fault.d         = vcc;

        ParInvertS.ena          = vcc;
        ParChopS.ena            = vcc;

        ParInvertS.d    = gnd;
        ParChopS.d      = gnd;
        Monitor         = AllOk;

WHEN Bus900 =>
        Fault.ena       = vcc;
        Fault.d         = vcc;

        AC_ACS.ena  = vcc;
        AC_ACS.d    = gnd;

        Monitor         = AllOK;

WHEN WaitS0 =>
        OldMonitor[].ena        = vcc;
        OldMonitor[].d          = Monitor.q[2..1];

        IF Reset.q THEN
                Monitor         = AllOff;
        ELSIF !Running.q THEN
                Monitor         = AllOff;
        ELSIF EOC AND !STC.q THEN
                Monitor         = EndW1;
        ELSE
                IF !EOC AND STC.q THEN
                        STC.ena         = vcc;
                        ALE.ena         = vcc;
                        STC.d = gnd;
                        ALE.d = gnd;
                END IF;
                Monitor         = SpeedR1;
        END IF;

    WHEN Waitbuses =>
        OldMonitor[].ena        = vcc;
        OldMonitor[].d          = Monitor.q[2..1];

        IF Reset.q THEN
                Monitor         = AllOff;
        ELSIF !EOC AND STC.q THEN
                STC.ena         = vcc;
                ALE.ena         = vcc;
```

```
                Speed/Torque.ena = vcc;
                STC.d          = gnd;
                ALE.d          = gnd;
                Speed/Torque.d       = !Speed/Torque.q;
            ELSIF EOC THEN
                Monitor              = EndW1;
            ELSIF 900VOn AND 800VOn THEN
                ParChopS.ena         = vcc;
                DCDriveS.ena         = vcc;
                SpeedS.ena           = vcc;
                TorqueS.ena          = vcc;

                ParChopS.d           = vcc;
                DCDriveS.d           = vcc;
                SpeedS.d             = vcc;
                TorqueS.d            = vcc;
                Monitor              = AllOk;
            END IF;

    WHEN AllOff =>
            ParInvertS.ena       = vcc;
            AC_ACS.ena           = vcc;
            ParChopS.ena         = vcc;
            DCDriveS.ena         = vcc;
            SoftStartS.ena       = vcc;
            DcDumpS.ena          = vcc;
            SpeedS.ena           = vcc;
            TorqueS.ena          = vcc;
            Fault.ena            = vcc;

            ParInvertS.d         = gnd;
            AC_ACS.d             = gnd;
            ParChopS.d           = gnd;
            DCDriveS.d           = gnd;
            SoftStartS.d         = gnd;
            DcDumpS.d            = vcc;
            SpeedS.d             = gnd;
            TorqueS.d            = gnd;
            Fault.d              = vcc;

            IF !Reset.q AND SysReset THEN
                Monitor         = SwitchOn;
            END IF;

    WHEN SwitchOn =>
            IF !Reset.q AND SysReset THEN
                Monitor         = AllOff;
            ELSIF Reset.q AND SysReset THEN
                Monitor         = SwitchOn;
```

```
            ELSE
                IF 4kVOn THEN
                    ParInvertS.ena          = vcc;
                    AC_ACS.ena              = vcc;
                    ParInvertS.d            = vcc;
                    AC_ACS.d                = vcc;
                    Monitor                 = WaitBuses;
                END IF;
                DcDumpS.ena         = vcc;
                SoftStartS.ena      = vcc;
                Fault.ena           = vcc;
                STC.ena             = vcc;
                DcDumpS.d           = gnd;
                SoftStartS.d        = vcc;
                Fault.d             = gnd;
                STC.d               = vcc;
            END IF;

        WHEN OTHERS =>
            Monitor = SwitchOn;

        END CASE;
END;
```